Path learning in human-robot collaboration tasks using iterative learning methods

Article (Accepted Version)

Path Learning in Human-Robot Collaboration Tasks Using Iterative Learning Methods

Xueyan Xing, Jingkang Xia, Deqing Huang, Yanan Li

***Abstract*—In a repetitive human-robot collaboration (HRC) task, robots typically are required to learn the intended motion of the human user to improve the collaboration efficiency. However, the human user's trajectory is of uncertainty when repeating a same task (e.g. human hand tremor and uncertain movement speed), which may directly deteriorate the learning performance. To address this issue, a path characterized by spatial correlation parameters, is of necessity to be learned by robots so that the aforementioned time-related uncertainty will be avoided. In this paper, based on the path parameterization, a gradient-based iterative path learning (IPL) strategy is designed for the robot to learn the desired path of human. The proposed IPL strategy draws on the iterative learning methods with a properly designed performance index. Since the gradient of the performance index is hard to directly obtain in HRC, two learning methods with gradient search (GS) and gradient estimation (GE) are developed. The GS estimates the gradient online and has an advantage of easy implementation. By contrast, the advantage of GS is more obvious when the number of learned parameters is considerable due to its high learning efficiency. With these two methods, the unknown path parameters can be iteratively updated towards the desired values. To verify the effectiveness of the proposed IPL algorithm, experiments are carried out. In the experiment, a comparison between GS and GE methods is made to display their respective advantages. Besides, the proposed IPL is compared with an existing trajectory learning method subject to two different kinds of uncertainties and its better learning performance verifies its stability and capability in dealing with uncertainty.**

***Index Terms*—Human-robot collaboration; path learning; iterative learning; gradient search; gradient estimation; path planning**

I. Introduction

At present, human-robot interaction is becoming a prevalent trend in factories, health care, and other applications, where robots are expected to cooperate with humans [1]–[3] and even compete with them [4]. Human-robot collaboration (HRC) is a field in human-robot interaction where robots need to collaborate with humans and usually fulfil physically demanding tasks [5], [6]. In order to successfully achieve the task goal in HRC, robots need to understand the intention of humans and act in accordance to it [7]. As a result, making robots efficiently and correctly infer the intention of humans, e.g. their desired trajectory, is an important research issue that needs to be addressed [8]. Different from [9] which always requires the human in the control loop, in this paper we focus on a scenario where after learning the robot is able to complete a task without the participation of human.

Machine learning methods have been widely used in HRC to enable the robot to learn the trajectory of the human user, where imitation learning (IL) and reinforcement learning (RL) are most common methods. IL, which is also known as programming by demonstration (PbD), is considered an intuitive method of robot programming. To implement PbD, the force and position information of the human needs to be collected. With the collected data, the demonstrated trajectory of the human can be modelled [10]–[12] such that the task geometry is not required in PbD. RL is a learning method that enables the robot to learn a certain desired behavior by repeatedly interacting with the environment [13]. It is extended to the field of HRC to make the robot learn the human user's uncertain intention by trial-and-error interaction [14]–[16]. Bayesian optimization [17] is served as an effective tool for minimizing experimental trials while optimizing a task and has been used for robot control in HRC. In [18], a hidden Markov model (HMM) is combined with Bayesian optimization, which has capability to learn a task with few demonstrations. In [19], Bayesian optimization is combined with PbD to make the robot learn the intended trajectory of human based on the noisy interaction force.

Compared with the machine learning method that is widely used in computer science, iterative learning is a concept originated in the control field [20], [21]. In the iterative learning method, learning can be achieved using more intuitive online information, such as output errors, in a repeatable task, instead of using probabilistic information as in the machine learning method. Besides this difference, the iterative learning method does not require multiple demonstrations for training. Since the research on iterative learning is relatively mature in the control field, its convergence can be proved using established control theory. In [22], a period-varying iterative learning strategy is proposed to teach the robot a desired trajectory of the human user in HRC. In [23], a trajectory learning approach is carried out so the robot can correct its trajectory in real time. Motion synchronization is realized in [24] using a force tracking framework and the robot can proactively follow the human with learning and neural networks control methods. Although the iterative learning method has above complementary advantages over the machine learning method, its application to HRC, especially to the trajectory learning, is still an open problem.

The work was supported by the UK EPSRC grant EP/T006951/1. (Corresponding author: Yanan Li.)

Xueyan Xing is with the Department of Engineering and Design, University of Sussex, Brighton BN1 9RH, UK. (email: xx92@sussex.ac.uk)

Yanan Li is with the Department of Engineering and Design, University of Sussex, Brighton BN1 9RH, UK. (email: yl557@sussex.ac.uk)

Jingkang Xia is with the School of Electrical Engineering, Southwest Jiaotong University, Chengdu, P. R. China. (email: xiajingkang@my.swjtu.edu.cn)

Deqing Huang is with the School of Electrical Engineering, Southwest Jiaotong University, Chengdu, P. R. China. (email: elehd@home.swjtu.edu.cn)

In aforementioned works [22]–[24], it is noteworthy that they all focus on trajectory learning. However, since the trajectory is time-related, the learning method works well only when the correct time mapping between iterations is guaranteed. As a result, the asynchronization between iterations may lead to wrong learning results. Moreover, the uncertainty related to time occurring during the learning process in HRC may also degrade the final learning performance of the robot. For instance, human may stop the task and guidance of the robot intentionally or due to external interruption. Then the robot's learning will be interrupted, eventually resulting in poor learning performance.

Compared with time-based trajectory, path is a concept related to space and has been widely studied in the field of HRC [25], [26]. The contour profile of path generally depends on path variables and path parameters. In the past decades, path planning methods have been developed in a variety of areas by optimizing performance indexes in an uncertain environment [27]–[30]. Path planning in these works is closely related to path learning for HRC, in the sense that the robot can ascertain the final reference path i.e. the human's desired path, by optimizing the performance index.

Regarding the tracking error in the established impedance control model as a performance index to optimize, the path learning problem of the robot in HRC is formulated. Two IPL methods are developed based on the inferred gradient information. By minimizing the performance index, the error between the robot's reference path and human's desired path can be iteratively reduced. The major contributions of this paper are elaborated as follows:

(1) In the traditional iterative learning method [20], the tracking error is needed as the feedback signal to achieve the control objective. However, in HRC, the human's desired path is unknown to the robot. To address this issue, a performance index including the interaction force is used as the feedback signal in this paper. As a result, the robot can effectively learn the human's desired path with our proposed iterative learning method.

(2) In previous works on trajectory learning [22]–[24], the robot learns the desired trajectory of the human user in a point-point manner in the time domain so that the control performance is sensitive to the time-related uncertainty. Different from these works, this paper minimizes the index performance in each iteration and updates the robot's reference path parameters. Since the path to learn is specified in the space domain instead of the time domain, the proposed method is inherently more robust against varying time durations and other uncertainties during learning.

(3) Different from the machine learning method that requires prior knowledge about the task model [17]–[19], the proposed method is suitable for tasks where the nominal plan of the robot does not conform with a presumed probabilistic model.

The main structure of this paper is illustrated in Fig. 1 and the remainder of this paper is organized as follows. Section II presents an impedance control model of HRC and converts a generic path learning problem to path learning for HRC. Section III shows two general gradient-based iterative path planning methods. According to the theoretical basis given in Section III, two IPL methods are developed in Section IV for HRC with the properly designed performance index function and the stability of the closed-loop system is analyzed. Experiment results are provided in Section V. Conclusions are finally presented in Section VI.

II. Problem Formulation

In this section, a generic problem of path planning is first formulated. Then an impedance control model is established for a robotic manipulator interacting with a human user. By matching the impedance control model with the general model, path planning is transformed into path learning for a typical HRC scenario.

A. Generic Problem of Path Planning

The problem of path planning is introduced in this subsection and the following general nonlinear model is considered [30]

$$\dot{\breve{\boldsymbol{x}}}(t) = f(\breve{\boldsymbol{x}}(t), \boldsymbol{u}(t)) \tag{1}$$

$$\boldsymbol{u}(t) = y(\breve{\boldsymbol{x}}(t), \boldsymbol{g}(\upsilon)) \tag{2}$$

where $\breve{\boldsymbol{x}}(t) \in R^N$ is the system state; $\boldsymbol{u}(t) \in R^M$ is the input signal of the system; $\boldsymbol{g}(\upsilon) \in R^3$ is a spatial vector and denotes the reference path which is known to the robot and needs to be optimized and followed by the control system.

In this paper, the path $\boldsymbol{g}(\upsilon)$ is parameterized and time-independent. It is related to the path variable $\upsilon \in R$ whose desired value $\upsilon^* \in R$ is unknown, $\upsilon \in [\underline{\upsilon}, \bar{\upsilon}]$, where $\underline{\upsilon}$ and $\bar{\upsilon}$ are two positive constants. Since only the closed path is discussed in this paper, $\underline{\upsilon}$ and $\bar{\upsilon}$ should be selected to satisfy $\boldsymbol{g}(\underline{\upsilon}) = \boldsymbol{g}(\bar{\upsilon})$.

According to the above model, it assumes that the low-level control $\boldsymbol{u}(t)$ in Eq. (2) has been fully designed and has the ability to stabilize the closed-loop system. Then the general form of the path planning problem can be described using the following performance index W and tracking condition

$$\min_{\boldsymbol{g}(\upsilon)} W(\boldsymbol{g}(\upsilon), \breve{\boldsymbol{x}}(0)) = \int_{\underline{T}}^{\bar{T}} \kappa(\breve{\boldsymbol{x}}(t), \boldsymbol{g}(\upsilon)) \mathrm{d}t \tag{3}$$

$$\text{s.t.: Eqs. (1) and (2)}$$

$$\min_{\upsilon} \{\mathrm{dist}(\boldsymbol{p}(t), \boldsymbol{g}(\upsilon))\} \leq \bar{d} \tag{4}$$

where $\boldsymbol{p}(t)$ is trajectory of the control system which can be measured using the position sensor; κ is a user-defined function related to the expected performance of the system; $\mathrm{dist}(\cdot,\cdot)$ denotes the spherical distance between the ends of the two input vectors and $\mathrm{dist}(\boldsymbol{p}(t), \boldsymbol{g}(\upsilon))$ is defined as the spherical distance between all points of the optimal path $\boldsymbol{g}(\upsilon)$ and $\boldsymbol{p}(t)$; $\bar{d}$ is a positive constant representing the maximum allowed distance tolerance in path following; $\breve{\boldsymbol{x}}(0)$ is the initial state of the system; $\underline{T}$ and $\bar{T}$ are the start and end times of the path planning process.

Eqs. (3) and (4) indicate the major objective of the path planning problem: an optimal path $\boldsymbol{g}(\upsilon) = \boldsymbol{g}^*(\upsilon)$ is expected to be found to minimize the performance index $W(\boldsymbol{g}(\upsilon), \breve{\boldsymbol{x}}(0))$, which is reflected by Eq. (3), under the condition that the system follows the prescribed path with

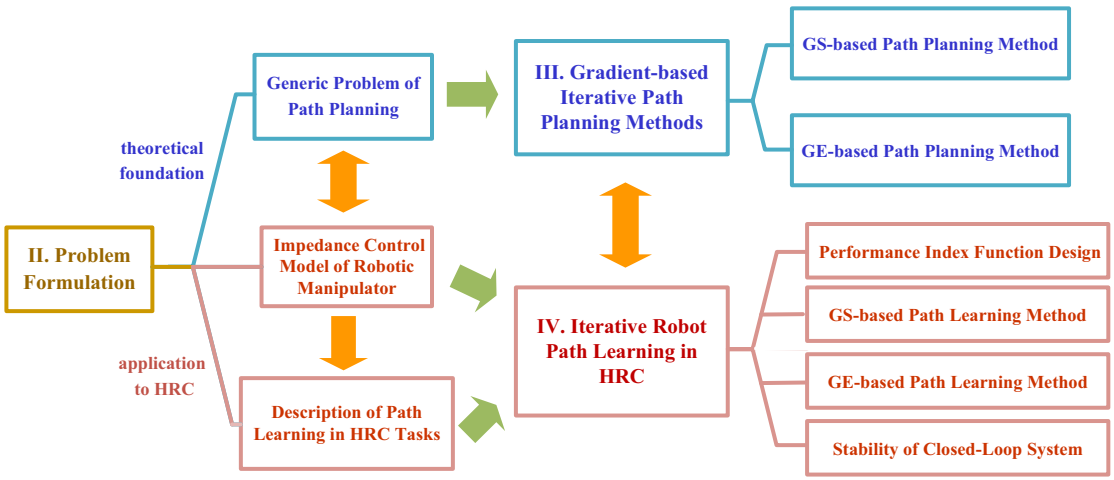


Fig. 1. Structure of this paper, where the light blue frame with dark blue text corresponds to the sections introducing the theoretical foundation and the sections represented by the pink frame with red text introduce the methods applying to path learning of HRC.

the allowable errors, which is reflected by Eq. (4). The performance index function $\kappa\left(\breve{\boldsymbol{x}}\left(t\right),\boldsymbol{g}\left(\upsilon\right)\right)$ will be designed in terms of practical needs.

To simplify the optimization condition, the constraint in Eq. (4) can be integrated into the performance index as a path-following penalty. Thus, the performance index in Eqs. (3) and (4) can be modified as

$$\begin{aligned}\min_{\boldsymbol{g}(\upsilon)} W\left(\boldsymbol{g}\left(\upsilon\right),\breve{\boldsymbol{x}}\left(0\right)\right)=&\int_{\underline{T}}^{\bar{T}}\kappa\left(\breve{\boldsymbol{x}}\left(t\right),\boldsymbol{g}\left(\upsilon\right)\right)\mathrm{d}t\\&+\frac{\varepsilon}{\bar{T}}\int_{\underline{T}}^{\bar{T}}h\left(\boldsymbol{p}\left(t\right),\boldsymbol{g}\left(\upsilon\right)\right)\mathrm{d}t\end{aligned}\tag{5}$$

where ε is a positive constant and $h\left(\boldsymbol{p}\left(t\right),\boldsymbol{g}\left(\upsilon\right)\right)$ is a nonnegative penalty function which contributes to find a sub-optimal solution when the optimal solution does not exist.

Remark 1: In the above path planning problem, it aims at finding an optimal path with υ^{*} which minimizes the performance index $W\left(\boldsymbol{g}\left(\upsilon\right),\breve{\boldsymbol{x}}\left(0\right)\right)$.

B. Impedance Control Model of Robotic Manipulator

In this subsection, the impedance control model of the robotic manipulator that is identical with the nonlinear system given by Eqs. (1) and (2) is established for HRC, which prepares for the subsequent gradient-based IPL method design.

The dynamics of the robotic manipulator with n-degrees-of-freedom (n-DOF) is described as

$$\begin{aligned}\boldsymbol{M}\left(\boldsymbol{q}\left(t\right)\right)\ddot{\boldsymbol{q}}\left(t\right)+\boldsymbol{C}\left(\boldsymbol{q}\left(t\right),\dot{\boldsymbol{q}}\left(t\right)\right)\dot{\boldsymbol{q}}\left(t\right)+\boldsymbol{H}\left(\dot{\boldsymbol{q}}\left(t\right)\right)\\+\boldsymbol{G}\left(\boldsymbol{q}\left(t\right)\right)=\boldsymbol{\tau}\left(t\right)+\boldsymbol{J}^{\mathrm{T}}\left(t\right)\boldsymbol{F}_{\boldsymbol{h}}\left(t\right)\end{aligned}\tag{6}$$

where $\boldsymbol{q}\left(t\right)\in R^{n}$ denotes the joint angle of the manipulator; $\boldsymbol{M}\left(\boldsymbol{q}\left(t\right)\right)\in R^{n\times n}$ stands for the symmetric positive definite mass matrix; $\boldsymbol{C}\left(\boldsymbol{q}\left(t\right),\dot{\boldsymbol{q}}\left(t\right)\right)\dot{\boldsymbol{q}}\left(t\right)\in R^{n\times n}$ is the torque caused by the Coriolis and centrifugal force; $\boldsymbol{G}\left(\boldsymbol{q}\left(t\right)\right)\in R^{n}$ represents the torque caused by the gravity; $\boldsymbol{H}\left(\dot{\boldsymbol{q}}\left(t\right)\right)\in R^{n}$ is a friction vector; $\boldsymbol{J}\left(t\right)\in R^{n\times n}$ denotes the Jacobian matrix; $\boldsymbol{\tau}\left(t\right)\in R^{n}$ is the input torque of the robot; $\boldsymbol{F}_{\boldsymbol{h}}\left(t\right)\in R^{n}$ is the interaction force which can be measured using a force sensor.

Denote the position/orientation of the end-effector of the manipulator in the Cartesian space as $\boldsymbol{x}\left(t\right)$ whose velocity and acceleration satisfy

$$\dot{\boldsymbol{x}}\left(t\right)=\boldsymbol{J}\left(t\right)\dot{\boldsymbol{q}}\left(t\right)\tag{7}$$

$$\ddot{\boldsymbol{x}}\left(t\right)=\boldsymbol{J}\left(t\right)\ddot{\boldsymbol{q}}\left(t\right)+\dot{\boldsymbol{J}}\left(t\right)\dot{\boldsymbol{q}}\left(t\right).\tag{8}$$

By combing Eqs. (6)-(8), the dynamics of the manipulator in the Cartesian space can be described as

$$\boldsymbol{M}\left(t\right)\ddot{\boldsymbol{x}}\left(t\right)+\boldsymbol{C}\left(t\right)\dot{\boldsymbol{x}}\left(t\right)+\boldsymbol{P}\left(t\right)=\boldsymbol{F}\left(t\right)+\boldsymbol{F}_{\boldsymbol{h}}\left(t\right)\tag{9}$$

where

$$\boldsymbol{M}\left(t\right)=\left(\boldsymbol{J}^{\mathrm{T}}\left(t\right)\right)^{-1}\boldsymbol{M}\left(\boldsymbol{q}\left(t\right)\right)\boldsymbol{J}^{-1}\left(t\right)\tag{10}$$

$$\begin{aligned}\boldsymbol{C}\left(t\right)=&\left(\boldsymbol{J}^{\mathrm{T}}\left(t\right)\right)^{-1}\left[\boldsymbol{C}\left(\boldsymbol{q}\left(t\right),\dot{\boldsymbol{q}}\left(t\right)\right)\right.\\&\left.-\boldsymbol{M}\left(\boldsymbol{q}\left(t\right)\right)\boldsymbol{J}^{-1}\left(t\right)\dot{\boldsymbol{J}}\left(t\right)\right]\boldsymbol{J}^{-1}\left(t\right)\end{aligned}\tag{11}$$

$$\boldsymbol{P}\left(t\right)=\left(\boldsymbol{J}^{\mathrm{T}}\left(t\right)\right)^{-1}\left(\boldsymbol{G}\left(\boldsymbol{q}\left(t\right)\right)+\boldsymbol{H}\left(\dot{\boldsymbol{q}}\left(t\right)\right)\right)\tag{12}$$

$$\boldsymbol{F}\left(t\right)=\left(\boldsymbol{J}^{\mathrm{T}}\left(t\right)\right)^{-1}\boldsymbol{\tau}\left(t\right).\tag{13}$$

We define the desired paths of the robot and human as $\boldsymbol{x}_{r}\left(\omega\right)\in R^{k}$ and $\boldsymbol{x}_{h}\left(\omega\right)\in R^{k}$, where $k\leq 3$ represents the dimension of the path; ω is a path variable defined on the ranges of $[\underline{\omega},\bar{\omega}]$ and $[\underline{\omega}',\bar{\omega}']$ for robot and human respectively; $\underline{\omega}$, $\bar{\omega}$, $\underline{\omega}'$, $\bar{\omega}'$ are four positive constants satisfying $\boldsymbol{x}_{\boldsymbol{r}}\left(\underline{\omega}\right)=\boldsymbol{x}_{\boldsymbol{r}}\left(\bar{\omega}\right)$ and $\boldsymbol{x}_{\boldsymbol{h}}\left(\underline{\omega}'\right)=\boldsymbol{x}_{\boldsymbol{h}}\left(\bar{\omega}'\right)$ to ensure the closure of $\boldsymbol{x}_{\boldsymbol{r}}\left(\omega\right)$ and $\boldsymbol{x}_{\boldsymbol{h}}\left(\omega\right)$. Then the path errors of the robotic manipulator and human are respectively defined as

$$\boldsymbol{e}\left(t\right)=\boldsymbol{x}\left(t\right)-\boldsymbol{x}_{\boldsymbol{r}}\left(\omega^{*}\right)\tag{14}$$

$$\boldsymbol{e}_{\boldsymbol{h}}\left(t\right)=\boldsymbol{x}\left(t\right)-\boldsymbol{x}_{\boldsymbol{h}}\left(\omega'\right)\tag{15}$$

where $\omega^{*}\in[\underline{\omega},\bar{\omega}]$ and $\omega'\in[\underline{\omega}',\bar{\omega}']$ are path variables which respectively make $\boldsymbol{x}_{\boldsymbol{r}}\left(\omega\right)$ and $\boldsymbol{x}_{\boldsymbol{h}}\left(\omega\right)$ be the closest points to the position $\boldsymbol{x}\left(t\right)$ of the control system at time t and $\boldsymbol{x}_{\boldsymbol{r}}\left(\omega^{*}\right)$ is known to the robot. Particularly, $\boldsymbol{x}_{\boldsymbol{h}}\left(\omega\right)$ and $\boldsymbol{e}_{\boldsymbol{h}}\left(t\right)$ are only

used for analysis and not used for control so they do not need to be measured.

Since $\boldsymbol{x}(t)$ varies with time, ω^* and ω' are also related to time and they can be respectively defined as

$$\omega^*(t) = \arg\min_{\omega\in[\underline{\omega},\bar{\omega}]} \{\text{dist}(\boldsymbol{x}(t), \boldsymbol{x_r}(\omega))\} \tag{16}$$

$$\omega'(t) = \arg\min_{\omega\in[\underline{\omega}',\bar{\omega}']} \{\text{dist}(\boldsymbol{x}(t), \boldsymbol{x_h}(\omega))\} \tag{17}$$

Then using Eq. (9), one obtains

$$\begin{aligned}&\boldsymbol{M_d}\ddot{\boldsymbol{e}}(t) + \check{\boldsymbol{C}}(t)\dot{\boldsymbol{e}}(t) + \check{\boldsymbol{P}}(t)\\ =&\boldsymbol{M_d}\boldsymbol{M}^{-1}(t)(\boldsymbol{F}(t) + \boldsymbol{F_h}(t))\end{aligned} \tag{18}$$

where $\boldsymbol{M_d}$ is a positive definite matrix, $\check{\boldsymbol{C}}(t) = \boldsymbol{M_d}\boldsymbol{M}^{-1}(t)\boldsymbol{C}(t)$, $\check{\boldsymbol{P}}(t) = \boldsymbol{M_d}\boldsymbol{M}^{-1}(t)(\boldsymbol{M}(t)\ddot{\boldsymbol{x}}_{\boldsymbol{r}}(\omega^*) + \boldsymbol{C}(t)\dot{\boldsymbol{x}}_{\boldsymbol{r}}(\omega^*) + \boldsymbol{P}(t))$, and $\dot{\boldsymbol{x}}_{\boldsymbol{r}}(\omega^*(t)) = \frac{d\boldsymbol{x_r}(\omega^*)}{d\omega^*}\frac{d\omega^*(t)}{dt}$.

In Eq. (18), the dynamics of the interaction force $\boldsymbol{F_h}(t)$ can be described using the following spring model [31]

$$\boldsymbol{F_h}(t) = \boldsymbol{k_h}(\boldsymbol{x_h}(\omega'(t)) - \boldsymbol{x}(t)) \tag{19}$$

where $\boldsymbol{k_h}$ is the stiffness matrix in the human force model.

In order to make the model in Eq. (18) more conforming to the general nonlinear model in Eq. (1), we design the low-level control as

$$\boldsymbol{F}(t) = \boldsymbol{M}(t)\boldsymbol{M_d}^{-1}\left[(\check{\boldsymbol{C}}(t) - \boldsymbol{k_d})\dot{\boldsymbol{e}}(t) - \boldsymbol{k_p}\boldsymbol{e}(t) + \check{\boldsymbol{P}}(t)\right] \tag{20}$$

where $\boldsymbol{k_d}$ and $\boldsymbol{k_p}$ are positive definite matrices that should be properly selected to guarantee the stability of the closed-loop system. Different from the classical impedance control in [32], the low-level control $\boldsymbol{F}(t)$ needs to be designed to make the closed-loop system have a general form as the one given in Eqs. (1) and (2).

Substituting Eq. (20) into Eq. (18) leads to

$$\boldsymbol{M_d}\ddot{\boldsymbol{e}}(t) + \boldsymbol{k_d}\dot{\boldsymbol{e}}(t) + \boldsymbol{k_p}\boldsymbol{e}(t) = \boldsymbol{M_d}\boldsymbol{M}^{-1}(t)\boldsymbol{F_h}(t). \tag{21}$$

Define $\boldsymbol{\xi}(t) = \begin{bmatrix}\boldsymbol{e}(t)\\ \dot{\boldsymbol{e}}(t)\end{bmatrix}$ and then the human's interaction force in Eq. (19) can be rewritten as

$$\boldsymbol{F_h}(t) = \boldsymbol{k_h}(\boldsymbol{x_h}(\omega'(t)) - \boldsymbol{x_r}(\omega^*(t))) - \boldsymbol{k_h}\boldsymbol{\Lambda}\boldsymbol{\xi}(t). \tag{22}$$

where $\boldsymbol{\Lambda} = \begin{bmatrix}1 & 0\end{bmatrix}$.

Eq. (21) can be written in the following form

$$\dot{\boldsymbol{\xi}}(t) = \boldsymbol{A'}(t)\boldsymbol{\xi}(t) + \boldsymbol{B'}(t)\boldsymbol{u'_e}(t) \tag{23}$$

where $\boldsymbol{A'}(t) = \begin{bmatrix}0 & \boldsymbol{I}\\ -\boldsymbol{M_d}^{-1}\boldsymbol{k_p} & -\boldsymbol{M_d}^{-1}\boldsymbol{k_d}\end{bmatrix}$, $\boldsymbol{B'}(t) = \begin{bmatrix}0\\ \boldsymbol{M}^{-1}(t)\end{bmatrix}$, $\boldsymbol{u'_e}(t) = \boldsymbol{F_h}(t)$.

Remark 2: The ways establishing the mapping between time and space to find ω^* and ω' in Eqs. (16) and (17) can be found in the researches of contouring control [33]–[35]. In the following experiments in Section V, the method in [35] with detailed steps is adopted to find ω^* and ω' at every instant.

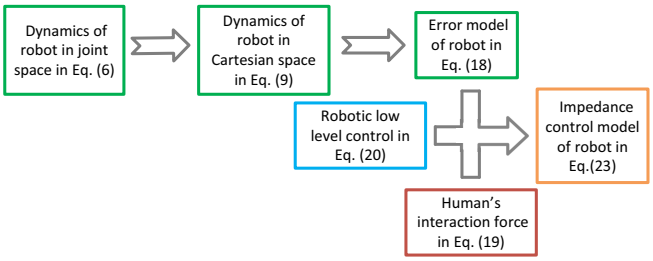


Fig. 2. Diagram of impedance control model in Section II-B and used for the experiment in Section V. The cross-arrow denotes that the impedance control model of robot in Eq. (23) is derived from the error model of robot in Eq. (18), the robotic low-level control in Eq. (20) and the human's interaction force in Eq. (19).

C. Description of Path Learning in HRC Tasks

In this section, a typical scenario of path learning in HRC tasks shown in Fig. 3 is considered. Different from path planning in Section II-A that only focuses on robot itself, the participation of human needs to be considered.

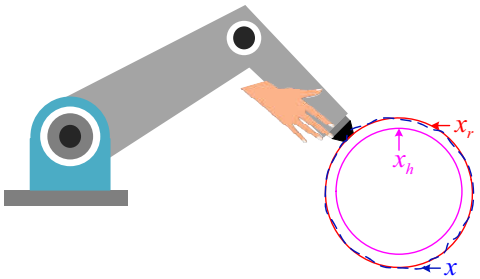


Fig. 3. Path learning process in HRC, where $\boldsymbol{x}$ is the actual path of the human-robot interaction system, $\boldsymbol{x_r}$ is the reference path of the robot, and $\boldsymbol{x_h}$ is the desired path of the human.

The robot path is represented using the basis parameter vector $\boldsymbol{\delta} \in R^s$, where s denotes the number of path parameters. Then $\boldsymbol{x_r}(\omega)$ in Eqs. (14), (16), (18), (22), and (23) is replaced by $\boldsymbol{x_r}(\omega, \boldsymbol{\delta})$. During HRC, $\boldsymbol{\delta}$ is expected to reach a desired value expected by the human and then the human's desired trajectory is learned by the robot, which can be realized using our following proposed IPL scheme after a few iterations.

In Fig. 3, the desired path of human depends on the practical needs. For instance, when the human wants the robot to help him/her to machine a workpiece, his/her desired path $\boldsymbol{x_h}(\omega)$ depends on the contour of the workpiece. In the path learning task, the human user holds the robot manipulator and guides it to follow $\boldsymbol{x_h}(\omega)$ which is known to him-/herself but unknown to the robot. Since $\boldsymbol{x_h}(\omega)$ is unknown to the robot, the robot needs to learn it and let $\boldsymbol{x_r}(\omega, \boldsymbol{\delta})$ converge to $\boldsymbol{x_h}(\omega)$ by iterations. To achieve the above objective, we need to transform the path planning problem in Section II-A into the path learning problem discussed in this paper.

Through observation, the transformed impedance control model of the robotic manipulator given by Eq. (23) can be

further expressed by

$$\dot{\boldsymbol{\xi}}(t) = \bar{f}(\boldsymbol{\xi}(t), \boldsymbol{u}'_{\boldsymbol{e}}(t)) \tag{24}$$

$$\boldsymbol{u}'_{\boldsymbol{e}}(t) = \bar{y}(\boldsymbol{\xi}(t), \boldsymbol{x}_{\boldsymbol{r}}(\omega, \boldsymbol{\delta})) \tag{25}$$

which is coherent with the general model in Eqs. (1) and (2).

As a result, the dynamics of the robotic manipulator can be described by the general model of the path planning problem in Section II-A and the path planning can be converted into a path learning problem in HRC with a corresponding path update law. It is noteworthy mentioning that we have specified the path in space rather than the trajectory in time, so the learning performance will not be affected by time-related uncertainties that will be further explained in Section IV-A.

III. Gradient-based Iterative Path Planning Methods

In this section, two gradient-based iterative path planning methods are developed based on the general nonlinear model in Eqs. (1)-(2) and the performance index in Eq. (5) in Section II-A.

The following two optimization schemes both aim at iteratively updating the parameterized path using the information of previous path parameters and corresponding performance index such that the updated path can ultimately converge to the optimum.

The path $\boldsymbol{g}(\upsilon)$ in Eq. (2) is considered to be parameterized and re-expressed by $\boldsymbol{g}(\upsilon, \boldsymbol{\sigma})$ where $\boldsymbol{\sigma} \in R^S$ is the basis path parameter vector. To optimize the path, the gradient information of the performance index function is generally needed, which is hard to obtain in some cases. Consequently, we can repeat a task to search for the unknown gradient or adopt adaptation laws to estimate the gradient.

A. Gradient Search (GS)-based Path Planning

In this part, GS is applied to path planning. By searching the descent direction of the gradient of the performance index function, the path can be optimized even if the gradient components are unknown.

According to the core idea of iterative learning, i.e. conducting a contractive mapping to realize iterative convergence, the iterative learning law of the path parameter vector is designed as follows [36]

$$\boldsymbol{\sigma}_{j+1} = \boldsymbol{\sigma}_j - \boldsymbol{\chi}_j W(\boldsymbol{\sigma}_j) \tag{26}$$

where the subscript j denotes the number of iterations; $\boldsymbol{\sigma}_j \in R^S$ represents the basis path parameter vector in the jth iteration; $\boldsymbol{\chi}_j \in R^S$ is a varying function that determines the convergence speed of iterations.

Define $\boldsymbol{\nabla} W(*)$ as the gradient of $W(*)$ at point $*$ and $\bar{\boldsymbol{\sigma}}_j \in [\min\{\boldsymbol{\sigma}_j, \boldsymbol{\sigma}_{j+1}\}, \max\{\boldsymbol{\sigma}_j, \boldsymbol{\sigma}_{j+1}\}]$. Then we have the following theorem.

Theorem 1: If the following condition is satisfied

$$\left|1 - \boldsymbol{L}_j^{\mathrm{T}} \boldsymbol{\chi}_j\right| \leq \rho < 1 \tag{27}$$

where $\boldsymbol{L}_j = [L_{1,j} \ \cdots \ L_{S,j}]^{\mathrm{T}} = \boldsymbol{\nabla} W(\bar{\boldsymbol{\sigma}}_j)$ and $\rho < 1$ is a positive constant, the minimization of the performance index function $W(\boldsymbol{\sigma}_j)$ can be guaranteed using the iterative updating law in Eq. (26).

The proof of Theorem 1 is given in Appendix 1.

Denote the path parameter vector and learning gain vector in the jth iteration as

$$\boldsymbol{\sigma}_j = [\sigma_{1,j} \ \cdots \ \sigma_{S,j}]^{\mathrm{T}} \tag{28}$$

$$\boldsymbol{\chi}_j = [\chi_{1,j} \ \cdots \ \chi_{S,j}]^{\mathrm{T}}. \tag{29}$$

According to Theorem 1, the following three cases, where $\boldsymbol{\chi}_j$ is designed to satisfy the condition in Eq. (27) mentioned by Theorem 1, are considered:

Case 1: If $\boldsymbol{L}_j$ is known, the learning gain $\boldsymbol{\chi}_j$ can be selected as $\chi_{i,j} = \frac{1}{SL_{i,j}}$, $i = 1, \cdots, S$, such that $\left|1 - \boldsymbol{L}_j^{\mathrm{T}} \boldsymbol{\chi}_j\right| = 0$ holds and the iterative learning law in Eq. (26) has the fastest learning convergence speed.

Case 2: If $\boldsymbol{L}_j$ is unknown, however, the boundary and sign information of $\boldsymbol{L}_j$ is known, we can still properly design $\chi_{i,j}$. For instance, if $0 < \underline{l}_i \leq L_{i,j} \leq \bar{l}_i$ holds where $\underline{l}_i$ and $\bar{l}_i$ are two positive known constants, $\chi_{i,j}$ can be designed as $\chi_{i,j} = \frac{1}{S\bar{l}_i}$ to satisfy $1 - \sum_{i=1}^{S} \frac{\underline{l}_i}{S\bar{l}_i} \leq \rho < 1$ such that Eq. (27) holds and the learning convergence is guaranteed.

Case 3: When $\boldsymbol{L}_j$ is unknown, GS is needed. First, the estimated gradient component $\hat{L}_{i,j}$ can be achieved from previous two iterations as follows [37]

$$\hat{L}_{i,j} = \frac{W(\boldsymbol{\sigma}_{j-1}) - W(\boldsymbol{\sigma}_{j-2})}{\sigma_{i,j-1} - \sigma_{i,j-2}}. \tag{30}$$

Then the magnitude of learning gain $\chi_{i,j}$ can be designed as

$$|\chi_{i,j}| = \varsigma_i \hat{L}_{i,j}^{-1} \tag{31}$$

in which $\varsigma_i \in (0, 1]$ is a positive constant to guarantee the validity of Eq. (27) in Theorem 1.

Since the gradient direction is unknown, the sign of the estimated gradient in Eq. (30) may be incorrect. Using the magnitude of learning gain $\chi_{i,j}$ in Eq. (31), $\chi_{i,j}$ is expressed by

$$\chi_{i,j} = \pm \varsigma_i \left| \frac{\sigma_{i,j-1} - \sigma_{i,j-2}}{W(\boldsymbol{\sigma}_{j-1}) - W(\boldsymbol{\sigma}_{j-2})} \right|. \tag{32}$$

As a result, extra learning trials are needed to be conducted to search for the correct sign of $\chi_{i,j}$ in Case 3. Once $\chi_{i,j}$ with correct sign is obtained, the condition in Eq. (27) is met and the learning method in Eq. (26) is effective.

Remark 3: The GS-based path planning method has a fixed form and few parameters so it is easy to implement. In this approach, due to the unknown sign of the gradient, the magnitude of the gradient is determined by Eq. (31) and then at least 2^S trials are needed in each searching round to find the descending direction of the performance index function.

B. Gradient Estimation (GE)-based Path Learning

Although the aforementioned GS-based approach has the advantage of easy implementation, extra trials are needed to determine the sign of the unknown gradient. To further increase the efficiency, the GE-based path learning method is

developed. In this approach, the performance index function is estimated by a recursive least square (RLS) based estimator and its gradient can be directly obtained.

The iterative adaptation law of path parameters is designed as [30]

$$\boldsymbol{\sigma}_{j+1} = \boldsymbol{\sigma}_j - \boldsymbol{K_J} \boldsymbol{\nabla} \hat{W}(\boldsymbol{\sigma}_j) \tag{33}$$

where $\boldsymbol{K_J} \in R^{S\times S}$ is a constant positive definite learning gain matrix; $\hat{W}$ is the estimated response surface of the performance index W; $\boldsymbol{\nabla}\hat{W}(\boldsymbol{\sigma}_j)$ denotes the gradient of $\hat{W}$ with current basis parameters $\boldsymbol{\sigma}_j$ in the jth iteration.

Remark 4: If the gradient of the performance index $W(\boldsymbol{\sigma}_j)$ is known, i.e. $\boldsymbol{\nabla}\hat{W}(\boldsymbol{\sigma}_j) = \boldsymbol{\nabla} W(\boldsymbol{\sigma}_j)$, the updating method in Eq. (33) can be regarded as the typical gradient descent (GD) method whose effectiveness has been proved by [30] and $\boldsymbol{K_J}$ is the step-size matrix. With accurate estimation of $\boldsymbol{\nabla} W(\boldsymbol{\sigma}_j)$, iterative decreasing of the performance index can be achieved using Eq. (33).

Since Eq. (33) is designed based on estimation of the gradient of the performance index $W(\boldsymbol{\sigma}_j)$, the following two steps are needed at each iteration before using Eq. (33) for path update:

Step 1. Estimating $W(\boldsymbol{\sigma}_j)$: To estimate $W(\boldsymbol{\sigma}_j)$, $\hat{W}(\boldsymbol{\sigma}_j)$ is modeled as follows:

$$\hat{W}(\boldsymbol{\sigma}_j) = \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j)\boldsymbol{\zeta}_j \tag{34}$$

where $\boldsymbol{\theta}(\boldsymbol{\sigma}_j) \in R^z$ is a regressor vector and $\boldsymbol{\zeta}_j \in R^z$ is a coefficient vector.

The coefficient vector $\boldsymbol{\zeta}_j$ in each iteration can be obtained using RLS as follows:

$$\boldsymbol{\vartheta}_j = \frac{1}{r}\left(\boldsymbol{\vartheta}_{j-1} - \frac{\boldsymbol{\vartheta}_{j-1}\boldsymbol{\theta}(\boldsymbol{\sigma}_j)\boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j)\boldsymbol{\vartheta}_{j-1}}{r + \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j)\boldsymbol{\vartheta}_{j-1}\boldsymbol{\theta}(\boldsymbol{\sigma}_j)}\right) \tag{35}$$

$$\boldsymbol{\zeta}_j = \boldsymbol{\zeta}_{j-1} + \frac{\boldsymbol{\vartheta}_j\boldsymbol{\theta}(\boldsymbol{\sigma}_j)}{r + \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j)\boldsymbol{\vartheta}_j\boldsymbol{\theta}(\boldsymbol{\sigma}_j)}\left(W(\boldsymbol{\sigma}_j) - \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j)\boldsymbol{\zeta}_{j-1}\right) \tag{36}$$

where $r \leq 1$ is a forgetting factor and positive semi-definite matrix $\boldsymbol{\vartheta}_j \in R^{z\times z}$ denotes the inverse of the weighted sample covariance matrix.

Remark 5: The structure of the regressor vector $\boldsymbol{\theta}(\boldsymbol{\sigma}_j)$ in Eq. (34) should be selected according to the expected functional relationship between $W(\boldsymbol{\sigma}_j)$ and path parameters $\boldsymbol{\sigma}_j$. If their functional relationship is implicit or uncertain, the regressor vector $\boldsymbol{\theta}(\boldsymbol{\sigma}_j)$ can be generally chosen as a set of polynomials as

$$\boldsymbol{\theta}(\boldsymbol{\sigma}_j) = \left[1 \ \sigma_{1,j} \ \cdots \ \sigma_{1,j}^{\gamma_1} \ \cdots \ \sigma_{S,j} \ \cdots \ \sigma_{S,j}^{\gamma_i}\right]^{\mathrm{T}} \tag{37}$$

in which γ_i is the maximum positive user-defined polynomial order.

By computing the inner product of $\boldsymbol{\theta}(\boldsymbol{\sigma}_j)$ and $\boldsymbol{\zeta}_j$, the estimate $\hat{W}(\boldsymbol{\sigma}_j)$ is obtained using Eq. (34) in the jth iteration.

Step 2. Calculating $\boldsymbol{\nabla}\hat{W}(\boldsymbol{\sigma}_j)$: In Eq. (34), the estimated response surface $\hat{W}(\boldsymbol{\sigma}_j)$ is represented by a function related to $\boldsymbol{\sigma}_j$ such that the gradient of $\hat{W}(\boldsymbol{\sigma}_j)$ can be computed as

$$\boldsymbol{\nabla}\hat{W}(\boldsymbol{\sigma}_j) = \left[\frac{\mathrm{d}\left(\boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j)\boldsymbol{\zeta}_j\right)}{\mathrm{d}\sigma_{1,j}} \ \cdots \ \frac{\mathrm{d}\left(\boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j)\boldsymbol{\zeta}_j\right)}{\mathrm{d}\sigma_{S,j}}\right]^{\mathrm{T}}. \tag{38}$$

Remark 6: Compared with the GS in Section III-A, the RLS-based estimator given by Eqs. (34)-(36) can effectively yield $\hat{W}(\boldsymbol{\sigma}_j)$ without extra trials, which greatly reduces the number of learning iterations, especially when the number of the path parameters is large. Besides, since $\boldsymbol{\nabla}\hat{W}$ can be directly calculated using the derivative of $\hat{W}$ with Eqs. (34) and (37), the complexity caused by differentiation is avoided.

In order to ensure the effectiveness of the RLS-based estimator in Eqs. (34)-(36), two assumptions are given as follows:

Assumption 1: The initial condition of the system $\breve{\boldsymbol{x}}(0)$ is independent of the iteration number, and the performance index can be parameterized as

$$W(\boldsymbol{\sigma}, \breve{\boldsymbol{x}}(0)) = W(\boldsymbol{\sigma}) = \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma})\boldsymbol{\zeta}^* \tag{39}$$

where $\boldsymbol{\zeta}^*$ is a constant vector.

Assumption 2: The performance index, $W(\boldsymbol{\sigma})$, is a convex and differentiable function and the gradient $\boldsymbol{\nabla} W(\boldsymbol{\sigma})$ is Lipschitz continuous with a positive constant α such that

$$\left\|\boldsymbol{\nabla} W(\boldsymbol{\sigma}_j^*) - \boldsymbol{\nabla} W(\boldsymbol{\sigma}_j)\right\| \leq \alpha \left\|\boldsymbol{\sigma}_{j+1} - \boldsymbol{\sigma}_j\right\| \tag{40}$$

where $\|\cdot\|$ denotes the Euclidian 2-norm and $\boldsymbol{\sigma}_j^* \in [\min\{\boldsymbol{\sigma}_j, \boldsymbol{\sigma}_{j+1}\}, \max\{\boldsymbol{\sigma}_j, \boldsymbol{\sigma}_{j+1}\}]$.

With Assumptions 1 and 2, the convergence of $\hat{W}(\boldsymbol{\sigma}_j)$ to $W(\boldsymbol{\sigma}_j)$ can be guaranteed using the gradient-based adaptive law in Eq. (33). The related theorem is given as follows.

Theorem 2: With Assumption 3 and the iterative adaptation law of path parameters in Eq. (33), if inequality $\psi_{\min}(\boldsymbol{K_J}) - \alpha\|\boldsymbol{K_J}\|^2 > 0$ is satisfied, we have $\lim_{j\to\infty}\left|\hat{W}(\boldsymbol{\sigma}_j) - W(\boldsymbol{\sigma}_j)\right| = 0$, where $\psi_{\min}(\boldsymbol{K_J})$ denotes the minimum eigenvalue of the learning gain matrix $\boldsymbol{K_J}$, and $W(\boldsymbol{\sigma}_j)$ converges and decreases during iterations.

The proof of Theorem 2 is given in Appendix 2.

Remark 7: In Theorem 2, Assumption 1 is used to guarantee that the performance index can be characterized by $\boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma})$. Assumption 2 ensures that the variation of the performance index is bounded during the iteration process and it is easy to be satisfied by selecting the performance index as a convex function. In some application scenarios, Assumption 1 may be hard to be satisfied due to the limitation of the geometrical shape of the path. In such a case, we can restrain the range of the basis path parameters between iterations, set the forgetting factor as $r < 1$, and decrease the learning gain $\boldsymbol{K_J}$ to enhance the robustness of the adaptive optimization law in Eq. (33).

IV. Iterative Robot Path Learning in HRC

In this section, two IPL methods are developed for the HRC on the basis of the gradient-based iterative path planning methods given in Section III.

The robot's spatial path in the jth iteration is given by

$$\boldsymbol{x}_{\boldsymbol{r},j}(\omega, \boldsymbol{\delta}_j) = [\phi_{1,j}(\boldsymbol{\delta}_j) \ \cdots \ \phi_{k,j}(\boldsymbol{\delta}_j)]^{\mathrm{T}} \tag{41}$$

where $\phi_{k,j}(\boldsymbol{\delta}_j)$ is a known shape function of parameter $\boldsymbol{\delta}_j$.

After properly designing the performance index function, the desired task path of the human user can be fully learned by the robot manipulator using the following developed path learning algorithms.

A. Performance Index Function Design

In this part, the performance index function is designed for HRC which lays the foundation for the subsequent design of path learning laws.

This paper aims at designing an algorithm to minimize the error between human's desired path and robot's reference path, which can be viewed as seeking an optimal basis path parameter vector $\boldsymbol{\delta} = \boldsymbol{\delta}^*$ to minimize the sum of $\boldsymbol{x_h}(\omega'(t)) - \boldsymbol{x_r}(\omega^*(t), \boldsymbol{\delta})$ in the path cycle. Observing Eq. (23), it is known that if $\boldsymbol{F_h}(t) = \mathbf{0}$ holds and properly select $\boldsymbol{k_d}$ and $\boldsymbol{k_p}$, $\boldsymbol{\xi}(t) = \mathbf{0}$ holds. Then according to Eq. (22), $\boldsymbol{x_h}(\omega'(t)) - \boldsymbol{x_r}(\omega^*(t), \boldsymbol{\delta}) = 0$ can be achieved. As a result, our final objective turns into

$$\min_{\delta} W = \min_{\delta} \int_0^T \|\boldsymbol{F_h}(t)\| \mathrm{d}t, \tag{42}$$

where T is the time period of one iteration, and the performance index used in the jth iteration is given as

$$W_j = \int_0^{T_j} \|\boldsymbol{F_h}_{,j}(t)\| \mathrm{d}t \tag{43}$$

where T_j is the time period of the jth iteration and the interaction force $\boldsymbol{F_h}_{,j}(t)$ in the jth iteration is obtained by letting $\boldsymbol{\delta} = \boldsymbol{\delta}_j$ in Eq. (23) so that W_j is a $\boldsymbol{\delta}_j$-related function.

During the minimization process of W, the path learning error can be reduced.

Remark 8: Due to the integral form of $\|\boldsymbol{F_h}(t)\|$ in Eq. (42), the performance index W can reflect the average path error of the robot, which reduces the undesirable effects caused by uncertainties during the path learning and improves the robustness. Using the path learning methods given below, the iteration period T_j, which can reflect the movement speed of human, is allowed to be uncertain and vary which is the case in practice.

Since the gradient of the performance index is hard to obtain during the path learning, the two methods given in Section III are used for the HRC scenario described in Section II-C.

B. GS-based Path Learning Method

The GS-based method in Section III-A is first considered in the IPL for HRC. Similar with Eqs. (26) and (30)-(32), the GS-based iterative law applied to the robot can be described as

$$\boldsymbol{\delta}_{j+1} = \boldsymbol{\delta}_j - \boldsymbol{\chi}_j W_j \tag{44}$$

where

$$\hat{L}_{i,j} = \frac{W_{j-1} - W_{j-2}}{\delta_{i,j-1} - \delta_{i,j-2}} \tag{45}$$

$$|\chi_{i,j}| = \varsigma_i \hat{L}_{i,j}^{-1} \tag{46}$$

$$\chi_{i,j} = \pm \varsigma_i \left| \frac{\delta_{i,j-1} - \delta_{i,j-2}}{W_{j-1} - W_{j-2}} \right| \tag{47}$$

and $i = 1, \cdots, s$.

The definition of ς_i and $\boldsymbol{\chi}_j$ are the same as the ones given in Section III-A and W_j is defined in Eq. (43). The sign of $\chi_{i,j}$ in Eq. (47) is unknown and extra path learning trials are required to identify it.

The block diagram of the path learning scheme with GS in Eqs. (44)-(47) is shown in Fig. 4.

To speed up the convergence and avoid getting stuck into local minima, the learning gain can be redesigned as ${\chi'}_{i,j}$ with a self-adaption rule [38], as below

$${\chi'}_{i,j} = \begin{cases} \chi_{i,j}\gamma, \text{ if } W(\delta_{i,j} - \chi_{i,j}\gamma) \leq W\left(\delta_{i,j} - \frac{\chi_{i,j}}{\gamma}\right) \\ \frac{\chi_{i,j}}{\gamma}, \text{ otherwise} \end{cases} \tag{48}$$

where γ is a variation factor to optimize the step ${\chi'}_{i,j}$ with the best overall fitting and $\chi_{i,j}$ is given in Eq. (47).

Then the iterative learning law of path parameters in Eq. (44) is revised as

$$\boldsymbol{\delta}_{j+1} = \boldsymbol{\delta}_j - {\boldsymbol{\chi}'}_j W_j \tag{49}$$

where $\boldsymbol{\chi}'_j = \left[\chi'_{1,j} \ \cdots \ \chi'_{s,j}\right]^{\mathrm{T}}$.

Using the scaled learning gain vector ${\boldsymbol{\chi}'}_j$, extra learning trials need to be conducted.

The major steps of GS-based path learning method for HRC can be summarized as: 1) Use Eq. (43) to calculate the performance index W_j in the jth iteration; 2) Calculate the preliminary convergence speed $\chi_{i,j}$ on the basis of Eqs. (45)-(47); 3) Obtain the optimized convergence speed ${\chi'}_{i,j}$ using Eq. (48); 4) Achieve the learned path parameter $\boldsymbol{\delta}_{j+1}$ using Eq. (49).

C. GE-based Path Learning Method

Based on the GE-based method in Section III-B, the adaptive path learning strategy using a RLS-based gradient estimator is proposed.

Using the following RLS-based estimator similar to Eqs. (35) and (36) in each iteration

$$\boldsymbol{\vartheta}_j = \frac{1}{r}\left(\boldsymbol{\vartheta}_{j-1} - \frac{\boldsymbol{\vartheta}_{j-1}\boldsymbol{\theta}(\boldsymbol{\delta}_j)\boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\delta}_j)\boldsymbol{\vartheta}_{j-1}}{r + \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\delta}_j)\boldsymbol{\vartheta}_{j-1}\boldsymbol{\theta}(\boldsymbol{\delta}_j)}\right) \tag{50}$$

$$\boldsymbol{\zeta}_j = \boldsymbol{\zeta}_{j-1} + \frac{\boldsymbol{\vartheta}_j\boldsymbol{\theta}(\boldsymbol{\delta}_j)}{r + \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\delta}_j)\boldsymbol{\vartheta}_j\boldsymbol{\theta}(\boldsymbol{\delta}_j)}\left(W_j - \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\delta}_j)\boldsymbol{\zeta}_{j-1}\right), \tag{51}$$

where the regressor vector is designed to contain the squares of path parameters as

$$\boldsymbol{\theta}(\boldsymbol{\delta}_j) = \left[1 \ \delta_{1,j} \ \delta_{1,j}^2 \ \cdots \ \delta_{s,j} \ \delta_{s,j}^2\right]^{\mathrm{T}}, \tag{52}$$

the estimation of the parameterized response surface is obtained as

$$\hat{W}_j = \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\delta}_j)\boldsymbol{\zeta}_j. \tag{53}$$

The definitions of r, $\boldsymbol{\vartheta}_j$ and $\boldsymbol{\zeta}_j$ can be found in Section III-B.

We rewrite $\boldsymbol{\zeta}_j$ in Eq. (51) as

$$\boldsymbol{\zeta}_j = \left[\zeta_{1,j}^{(0)} \ \zeta_{1,j}^{(1)} \ \zeta_{1,j}^{(2)} \ \cdots \ \zeta_{s,j}^{(0)} \ \zeta_{s,j}^{(1)} \ \zeta_{s,j}^{(2)}\right]^{\mathrm{T}} \tag{54}$$

and the gradient of $\hat{W}_j$ can be directly calculated as

$$\boldsymbol{\nabla}\hat{W}_j = \left[\zeta_{1,j}^{(1)} + 2\zeta_{1,j}^{(2)}\delta_{1,j} \ \cdots \ \zeta_{s,j}^{(1)} + 2\zeta_{s,j}^{(2)}\delta_{s,j}\right]^{\mathrm{T}}. \tag{55}$$

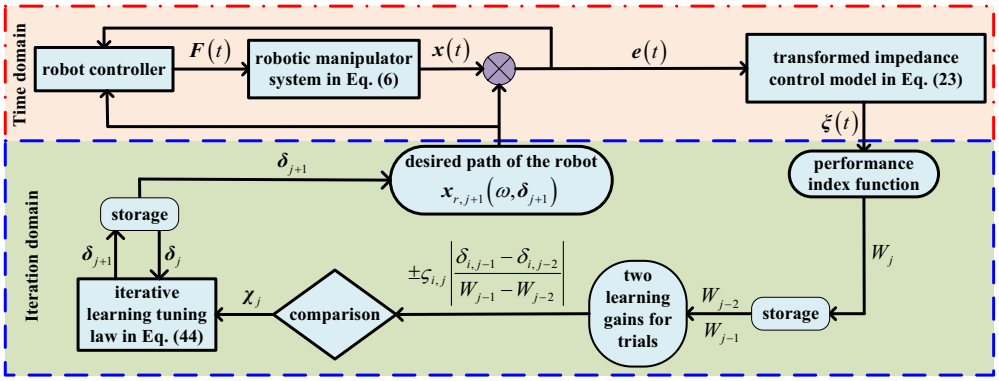


Fig. 4. Block diagram of the GS-based path learning approach.

According to Eq. (55), the path adaptation law is designed as

$$\boldsymbol{\delta}_{j+1} = \boldsymbol{\delta}_j - \boldsymbol{K_J}\left(\nabla\hat{W}_j, \Delta\boldsymbol{\delta_{max}}\right)\nabla\hat{W}_j \tag{56}$$

$$\boldsymbol{K_J}\left(\nabla\hat{W}_j, \Delta\boldsymbol{\delta_{max}}\right) = \min\left\{1 \quad \frac{\Delta\delta_{1,\max}}{\left|\zeta_{1,j}^{(1)} + 2\zeta_{1,j}^{(2)}\delta_{1,j}\right|} \quad \cdots \right.$$
$$\left. \frac{\Delta\delta_{s,\max}}{\left|\zeta_{s,j}^{(1)} + 2\zeta_{s,j}^{(2)}\delta_{s,j}\right|}\right\}\boldsymbol{\nu} \tag{57}$$

where $\boldsymbol{\nu}$ is a positive definite diagonal matrix with proper dimensions, and the learning gain $\boldsymbol{K_J}$ in Eq. (33) is revised to be related to $\nabla\hat{W}_j$ and a user-defined constant trust region $\Delta\boldsymbol{\delta_{max}} = [\Delta\delta_{1,\max} \ \cdots \ \Delta\delta_{s,\max}]^{\mathrm{T}} \in R^s$.

From Eq. (57), it is known that the revised learning gain $\boldsymbol{K_J}$ varies from iteration to iteration such that the adaptability and convergence of the proposed strategy in Eq. (56) can be improved. Besides, we can enhance the robustness of the optimization by adjusting $\Delta\boldsymbol{\delta_{max}}$ in Eq. (57).

The block diagram of the path adaptation strategy with the RLS-based gradient estimator in Eqs. (50)-(57) is shown in Fig. 5.

The path adaptation law in Eq. (56) can be further improved by using the Adam method [39] which can be regarded as an improved GD method to avoid the local optimum. Then Eq. (56) can be revised as follows

$$\boldsymbol{\beta}_j = \alpha_1\boldsymbol{\beta}_{j-1} + (1-\alpha_1)\nabla\hat{W}_j \tag{58}$$

$$\mu_j = \alpha_2\mu_{j-1} + (1-\alpha_2)\nabla\hat{W}_j^T\nabla\hat{W}_j \tag{59}$$

$$\hat{\boldsymbol{\beta}}_j = \frac{\boldsymbol{\beta}_j}{1-\alpha_1} \tag{60}$$

$$\hat{\mu}_j = \frac{\mu_j}{1-\alpha_2} \tag{61}$$

$$\boldsymbol{\delta}_{j+1} = \boldsymbol{\delta}_j - \frac{\boldsymbol{K_J}\left(\nabla\hat{W}_j, \Delta\boldsymbol{\delta_{max}}\right)}{\sqrt{\hat{\mu}_j+\varepsilon}}\hat{\boldsymbol{\beta}}_j \tag{62}$$

where α_1 and α_2 are positive hyper-parameters determining the exponential decay rates of vector $\hat{\boldsymbol{\beta}}_j$ and $\hat{\mu}_j$, and ε is a positive constant to avoid the possible singularity.

The major steps of GE-based path learning method for HRC can be summarized as: 1) Employ Eq. (43) to calculate the performance index W_j in the jth iteration; 2) Obtain the parameterized response surface $\hat{W}_j$ using Eq. (53) and the RLS-based estimator defined in Eqs. (50)-(53); 3) On the basis of Step 2, $\nabla\hat{W}_j$ can be calculated with Eq. (55); 4) Achieve the learned path parameter $\boldsymbol{\delta}_{j+1}$ according to the path adaptation law given in Eqs. (56) and (57).

Remark 9: The GE-based path learning approach given in Section IV-C can effectively calculate $\nabla\hat{W}_j$ without any extra trials and thus it is more efficient and suitable for path learning in HRC tasks when multiple parameters need to be estimated. However, since its structure is more complicated and has plenty of parameters, the GE-based path learning method is hard to implement compared with the GS-based path learning strategy in Section IV-B. When the number of uncertain path parameters is small, the GS-based path learning method is recommended.

D. Stability of Closed-Loop System

With the GS-based path learning method in Section IV-B and the GE-based path learning method in Section IV-C, the stability of the human-robot interaction system can be guaranteed, as summarized in the following theorem.

Theorem 3: With the developed GS-based/GE-based path learning method and the robotic controller in Eq. (20), the closed-loop system is asymptotically stable in the sense of $\lim_{j\to\infty}\boldsymbol{e}_j(t) = 0$ and $\lim_{j\to\infty}\dot{\boldsymbol{e}}_j(t) = 0$.

The proof of Theorem 3 can be found in Appendix 3.

V. Experiment

This section aims at evaluating the proposed IPL method by performing comparative experiments.

A. Experimental Setup

As shown in Fig. 6(a), the 2-DOF robotic platform H-MAN that is exclusively designed for human-robot interaction is equipped with an ATI Mini-40 force/torque sensor on its handle and position sensor with high precision. Human user and the robot are connected by the movable handle so human can provide guidance to the robot in path learning through the interaction force. The 7-DoF Sawyer robot which is more suitable for industrial applications is shown in Fig. 6(b). Robotiq

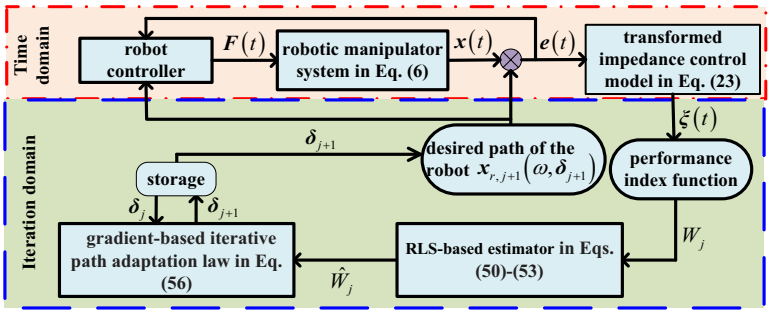
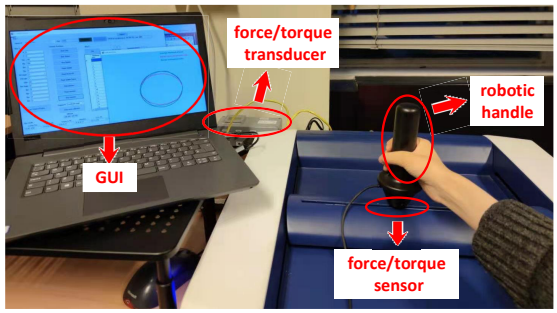
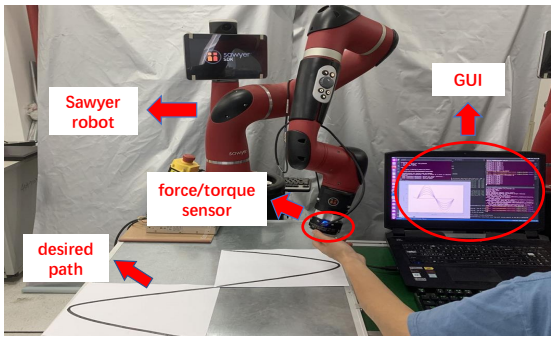


Fig. 5. Block diagram of the GE-based path learning approach.

FT300 torque/force and position sensors are equipped with motor encoders on each joint. In experiments, the real-time position is required to transmit through the communication links from H-MAN/Sawyer robot to the control computer as control signals, and the graphical user interface (GUI) including graphics of $\boldsymbol{x}$, $\boldsymbol{x_h}$ and $\boldsymbol{x_r}$ is displayed on a monitor.



(a)



(b)

Fig. 6. Experimental setup. (a) Platform – H-MAN robot (b) Platform – Sawyer robot.

For experiments in Section V-B,C,D, the human's desired path is set as an ellipse that is defined as $\boldsymbol{x_h}(\omega) = \begin{bmatrix} 0.8\cos\pi\omega & 0.7\sin\pi\omega \end{bmatrix}^{\mathrm{T}}$ with start point $\begin{bmatrix} 0.8 & 0 \end{bmatrix}^{\mathrm{T}}$. The robot's iterative path is given as $\boldsymbol{x_{r,j}}(\omega, \boldsymbol{\delta}_j) = \begin{bmatrix} 0.8\cos\pi\omega & \delta_{1,j}\sin\pi\omega \end{bmatrix}^{\mathrm{T}}$, where $\boldsymbol{\delta}_j = \delta_{1,j}$ is the basis path parameter to be iteratively updated by the robot during interaction and its initial value is set as $\boldsymbol{\delta}_0 = 0.55$. Parameters of the low-level controller are chosen as $\boldsymbol{M_d} = 1$, $\boldsymbol{k_d} = 200$ and $\boldsymbol{k_p} = 1500$. The control inputs of the human-robot interaction system are designed as in Fig. 2.

B. Comparison Between GS and GE-based Path Learning Methods for 1D Path with H-MAN Robot

The learning parameters of GS and GE-based path learning methods are respectively designed as $\varsigma_1 = 0.12$, and $r = 0.03$, $\boldsymbol{\nu} = 0.06$, $\alpha_1 = 0.2$, $\alpha_2 = 0.2$, $\Delta\boldsymbol{\delta}_{\mathbf{max}} = 1$, $\varepsilon = 0.001$. With these parameters, the experimental results of the path of the robotic system, the updated path parameter, the performance index are presented in Fig. 7 for the GS-based method and in Fig. 8 for the GE-based method, where $\boldsymbol{x}_j$ denotes the actual position of the robot in the jth iteration.

Figs. 7(a) and 8(a) illustrate the desired path of the human $\boldsymbol{x_h}$, the reference path of the robot $\boldsymbol{x_{r,j}}$ in different iterations, and the actual path of the robot. The updated path parameter $\boldsymbol{\delta}_j$ and the performance index W_j under two path learning methods are respectively reflected by Figs. 7(b) and 8(b). The self-adaption rule in Eq. (48) is adopted in the GS-based method in the experiment and the four trials in every iteration of GS-based scheme are shown in Table I, in which the first two trials are used to determine the sign of the gradient in Eq. (47) and the last two trials are used for self-adaption. From Table I, it can be seen that the robot's path parameter is updated in every trial and the one that best suits the need of the user can be selected as the path parameter in the next iteration by searching. In GS-based method, the performance index W_j is set as the minimal one in four trials.

From Figs. 7 and 8, it is obvious that the reference path of the robot coincides with the human's desired path after learning and the robot can effectively learn the human's desired path under both methods. In the two proposed IPL schemes, the performance index gradually decreases and finally converges to the neighborhood of zero.

When there is one uncertain path parameter, the GS-based path learning method needs 3 iterations and the GE-based one requires 8 iterations, which illustrates that when the uncertain path parameters are of few number, GS has the advantage of simpler structure and fewer learning parameters. When the number of uncertain path parameters increases, extra trials of GS-based path learning method become nontrivial and the path learning method should be reselected in the light of user requirements.

To compare the performance of two path learning approaches, we repeat the experiment five times for each approach under the same condition. The final reference path parameter learned by the robot is recorded and shown in Fig. 9. Ac-

TABLE I: Path parameters learned by robot with GS

j	δ_j	four trials in the jth iteration			
		$1st\ trial$	$2nd\ trial$	$3rd\ trial$	$4th\ trial$
0	0.5500	0.5500	0.5500	0.5500	0.5500
1	0.5500	0.6160	0.5500	0.5500	0.5500
1	0.5500	0.6160	0.4840	0.5500	0.5500
1	0.5500	0.6160	0.4840	0.6714	0.5500
1	0.5500	0.6160	0.4840	0.6714	0.5859
2	0.6714	0.6877	0.4840	0.6714	0.5859
2	0.6714	0.6877	0.6500	0.6714	0.5859
2	0.6714	0.6877	0.6500	0.7014	0.5859
2	0.6714	0.6877	0.6500	0.7014	0.6802
3	0.7014	0.7014	0.6500	0.7014	0.6802
3	0.7014	0.7014	0.7007	0.7014	0.6802
3	0.7014	0.7014	0.7007	0.7014	0.6802
3	0.7014	0.7014	0.7007	0.7014	0.7014

cording to these, the GS and GE-based path learning methods have similar performance with an error less than $0.01\ m$, which demonstrates that the two proposed methods both have favourable repeatability.

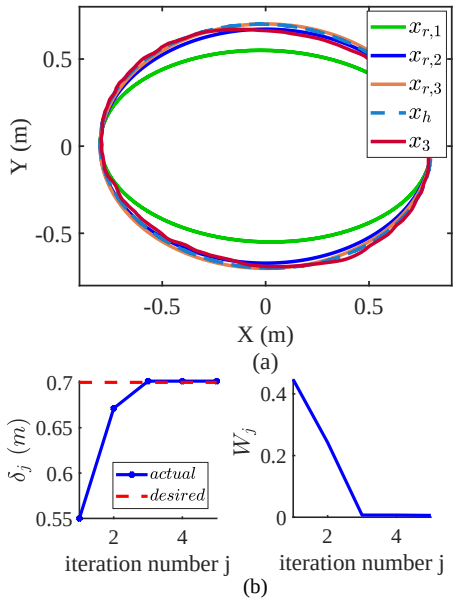


Fig. 7. Reference and actual paths of the H-MAN robot and updated path parameter in every iteration under the GS-based path learning method in Section V-B. (a) reference and actual paths of the robot (b) updated path parameter and performance index value.

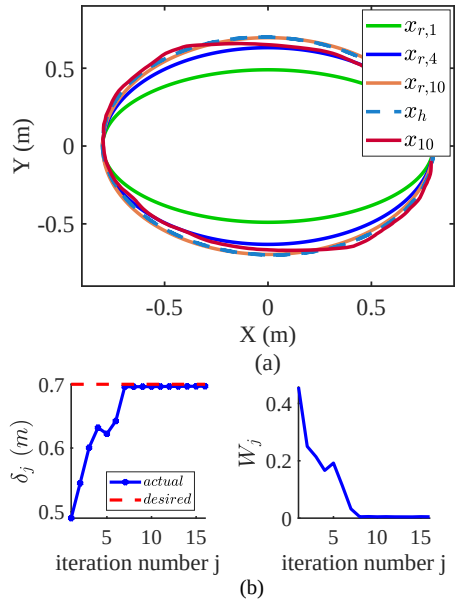


Fig. 8. Reference and actual paths of the H-MAN robot and updated path parameter in every iteration under the GE-based path learning method in Section V-B. (a) reference and actual paths of the robot (b) updated path parameter and performance index value.

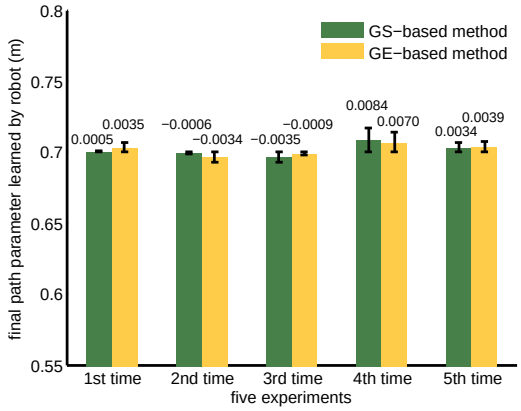


Fig. 9. Final path parameters learned by the H-MAN robot in five reduplicate experiments using GS and GE-based path learning methods. The black line segments represent the degree of the standard deviation from the average value.

C. Comparison with Existing Trajectory Learning Method for 1D Path with H-MAN Robot

To further verify the capability of the proposed IPL method in dealing with uncertainties, the GE-based path learning method in Section V-B is compared with the existing trajectory learning method in [22] with initial trajectory $\boldsymbol{x}_{\boldsymbol{r},0}\left(t\right)=\left[0.8\cos\pi t \quad 0.55\sin\pi t\right]^{\mathrm{T}}$ and learning parameters $\boldsymbol{\beta}_{\boldsymbol{r}}=2$.

The iterative trajectory learning method in [22] can be summarized as

$$\boldsymbol{x}_{\boldsymbol{r},j+1}\left(t\right)=\boldsymbol{x}_{\boldsymbol{r},j}\left(t\right)+\boldsymbol{\beta}_{\boldsymbol{r}}\boldsymbol{F}_{\boldsymbol{h}}\left(t\right) \tag{63}$$

where $\boldsymbol{\beta}_{\boldsymbol{r}}$ is the parameter matrix of iterative learning and $\boldsymbol{F}_{\boldsymbol{h}}\left(t\right)$ is the interaction force in real time.

Then the following two kinds of uncertainties are considered:

Case 1: The hand tremor of the human user occurs during the path learning;

Case 2: The human user prematurely stops his guidance at around a fifth of the path during the path learning.

It is noteworthy that the uncertainties mentioned in cases 1 and 2 occur in every iteration period with a similar pattern – similar amplitude and frequency of hand tremor in case 1 and similar stop position in case 2.

Since the GS and GE-based methods have similar performance, we used the latter one for the following comparison. The actual path under the trajectory learning method in [22] and the proposed GE-based path learning method in cases 1 and 2 are respectively presented in Figs. 10 and 11. According

TABLE II: Comparison made in experiment in Section V

	Section V-B		Section V-C	
Comparison	Proposed GS-based and GE-based methods without uncertainties		Existing trajectory learning in [21] and proposed IPL methods with uncertainties	
Method	Proposed GS-based method	Proposed GE-based method	Existing trajectory learning method in [21]	Proposed IPL method
Property	With extra trials and less learning parameters	Fast learning without extra trials	Sensitive to uncertainties and unstable	Stable and robust against uncertainties
Conclusion	GS is easy to implement and suits for the path with fewer parameters. In contrast, GE has faster learning and suits for the path with more parameters.		Proposed IPL method has better learning performance than the existing trajectory learning method in [21] in presence of various uncertainties.	

to Fig. 10(a), it is found that the hand tremor in case 1 is "recorded" by the existing trajectory learning method and the reference trajectory of robot suffers from obvious buffeting that results in the undesirable vibration and degrades the control performance of the system. In comparison, in Fig. 10(b), the tremor is filtered by the the proposed path learning method due to the integral form of the performance index W_j and the desired path parameter can be correctly learned by the robot.

Since the learning method in [22] updates the robot's reference trajectory point to point, only part of the reference trajectory can be learned by the robot in case 2. Once the human user stops applying the interaction force and stops his/her guidance (the stop point is marked in Fig. 11(a)), the robot immediately stops its learning and the learned reference trajectory is thus incomplete in Fig. 11(a). Compared with trajectory learning, the path learning method proposed in this paper can still obtain the complete path as shown in Fig. 11(b), even if the human only guides the robot through part of the path in every iteration. These results illustrate a great advantage of the proposed IPL method in the sense that the user only needs to guide the robot through part of the path so the learning efficiency is improved.

Then we repeatedly conduct the experiment five times with varying degrees of hand tremor for case 1 and premature termination for case 2, respectively, under the same condition. For each case, the GE-based path learning method is applied and 15 iterations are conducted. Define the average interaction force as

$$\bar{F}_{h,j} = \frac{\|\boldsymbol{F}_{\boldsymbol{h},j}(t)\|}{T_j}. \tag{64}$$

The final learned path parameter and the average interaction force $\bar{F}_{h,j}$ are respectively shown in Figs. 12(a) and 12(b). From Fig. 12(a), it is observed that the proposed IPL method can repetitively achieve similar performance in cases 1 and 2 with errors $0.7 - \boldsymbol{\delta}_j$ less than $0.01\ m$ when the iteration is convergent. In Fig. 12(b), the average interaction force converges to zero with iteration in both cases 1 and 2, which indicates that the robot gradually changes its reference trajectory towards the desired one of the human user so he/she finally terminates exerting interaction force. The increasing interaction force in Fig. 12(b) is caused by the local optimum. Once the robot gets stuck at a locally optimal value before obtaining the desired path parameter, users should properly increase their force. Besides, Fig 12(b) reveals that for case 2 where human stops the guidance prematurely, the number of iterations is slightly increased to ensure the good performance.

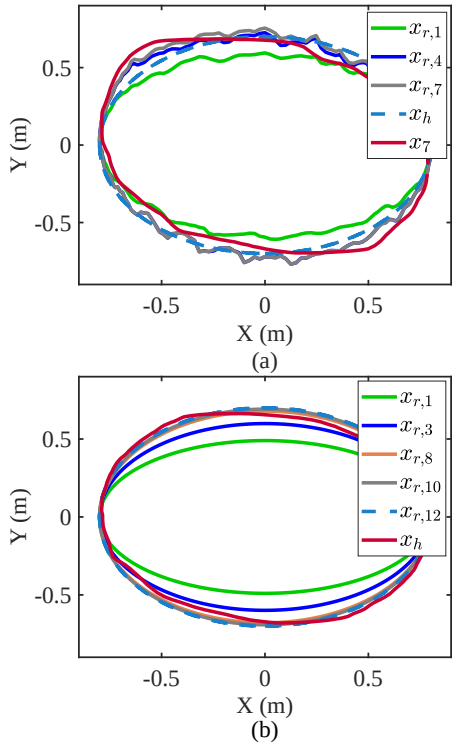


Fig. 10. Reference and actual paths of the H-MAN robot in every iteration under (a) the trajectory learning method in [22] and (b) the GE-based path learning method in case 1.

With the proposed IPL method, the interaction force is iteratively reduced and the error of the path parameter converges to a small neighborhood region of zero for five reduplicate experiments, which demonstrates its effectiveness and convergence. Comparisons made in experiment in Section V are summarized in Table II.

D. 2D Path Learning with H-MAN Robot

To test the generalizability and capability of learning path with multiple parameters of the proposed algorithm, we further test the H-MAN robot to learn a 2D elliptical path with two parameters using the GE-based method.

The initial path of the robot and the desired path of the human are respectively set as $\boldsymbol{x}_{\boldsymbol{r},0}(t) = \begin{bmatrix} 0.3\cos\pi t & 0.8\sin\pi t \end{bmatrix}^{\mathrm{T}}$ and $\boldsymbol{x}_{\boldsymbol{h}}(t) = \begin{bmatrix} 0.6\cos\pi t & 0.6\sin\pi t \end{bmatrix}^{\mathrm{T}}$. Parameters of the low-level controller are chosen as $\boldsymbol{M_d} = \mathrm{diag}(1,1)$, $\boldsymbol{k_d} = \mathrm{diag}(200,200)$ and $\boldsymbol{k_p} = \mathrm{diag}(1500,1500)$. The learning parameters of the GE-based method are designed as $r = 0.03$,

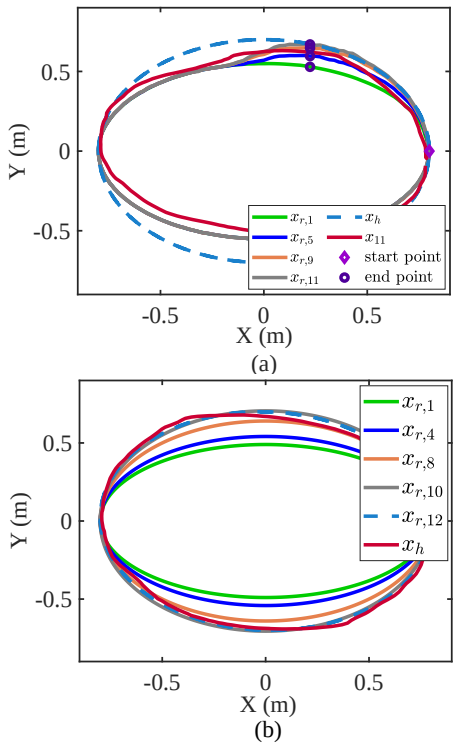


Fig. 11. Reference and actual paths of the H-MAN robot in every iteration under (a) the trajectory learning method in [22] and (b) the GE-based path learning method in case 2.

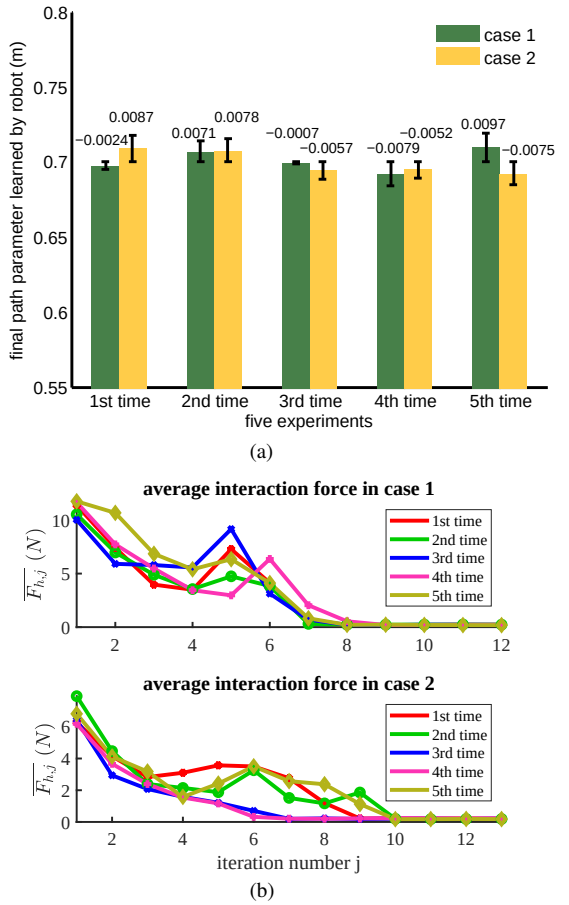


Fig. 12. Final path parameter learned by the H-MAN robot (a) and average interaction force (b) in five reduplicate experiments under the GE-based path learning method in cases 1 and 2. The black line segments in (a) represent the degree of the standard deviation from the average value.

$\boldsymbol{\nu} = \text{diag}(0.06, 0.06)$, $\alpha_1 = \alpha_2 = 0.2$, $\varepsilon = 10^{-3}$, and $\Delta\boldsymbol{\delta}_{\mathbf{max}} = \begin{bmatrix}1 & 1 & 1\end{bmatrix}^{\mathrm{T}}$.

The learning results of the 2D elliptical path are revealed by Fig. 13, where the robot learns the desired path of human within 10 iterations and both learning accuracy and speed are similar with that in Section V-B. Consequently, the increase of the number of learned parameters does not affect the overall learning performance of our proposed algorithm, which demonstrates its generalizability in learning various paths.

E. Path Learning with Sawyer Robot

Finally, Sawyer robot is used to verify the feasibility of the proposed path learning method using the GS-based method (visit website https://youtu.be/UJc7nzZAaGE to see the video with the experimental results).

The initial path parameter of the Sawyer robot is designed as $\boldsymbol{\delta}_0 = 0.55$ and the desired path of human is set by $\boldsymbol{x_h}(t) = 0.3\cos\pi t$. Parameters of the low-level controller are set as $\boldsymbol{M_d} = 1$, $\boldsymbol{k_d} = 200$ and $\boldsymbol{k_p} = 1500$. The learning parameters of the GE-based method are designed as $r = 0.03$, $\boldsymbol{\nu} = 0.06$, $\alpha_1 = \alpha_2 = 0.2$, $\varepsilon = 10^{-3}$, and $\Delta\boldsymbol{\delta}_{\mathbf{max}} = 1$.

As illustrated by Fig. 14, the learned parameter converges to the desired value after 7 iterations and the value of the performance index function gradually decreases to zero as expected. The experiment results in Fig. 14 show that the proposed path learning strategy in this paper is applicable to different types of robots.

VI. Conclusions

This paper presents two IPL methods for HRC tasks, using GS and GE to obtain the uncertain gradient of the performance index. The GS-based path learning method can be regarded as an online method with extra trials to improve the performance of searching path parameters. Compared with the GS-based method that has fewer learning parameters and is easier to be implemented, the GE-based path learning method allows to learn the path parameters using a RLS-based estimator without extra trials, so it is a better alternative when the human's desired path needs to be described with more parameters. Using the developed IPL methods, the robot can effectively learn the human's desired path by iteratively adjusting its path parameters and updating its path with minimizing the designed performance index. Compared with the existing point-to-point trajectory learning, since the paths of both human and robot are time independent and the performance index is in the form of integral on every iteration period, the IPL methods are robust against time-correlated uncertainties, as demonstrated by simulations and experiments. Despite aforementioned advantages, the IPL methods are only applicable to the path whose parameters are unknown but geometry is known. In the future work, we plan to combine the proposed IPL methods

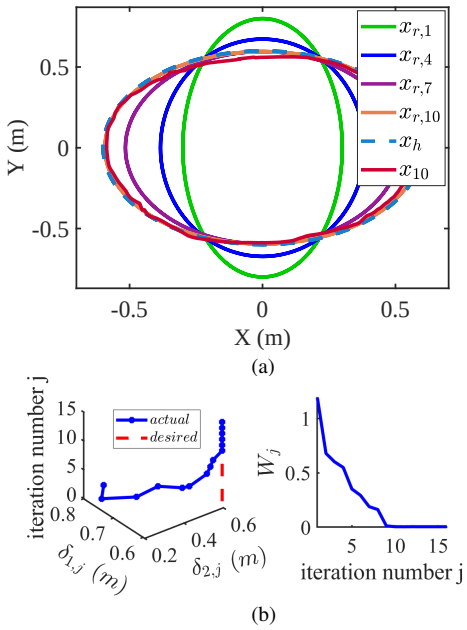


Fig. 13. Reference and actual paths of the H-MAN robot and updated path parameters in every iteration under the GE-based path learning method in Section V-D. (a) reference and actual paths of the robot (b) updated path parameters and performance index value.

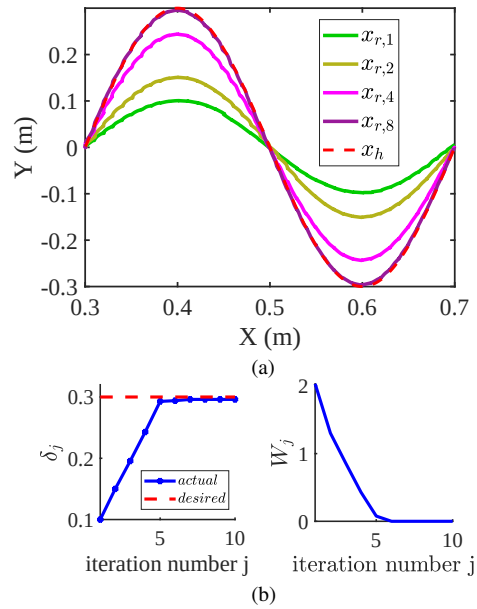


Fig. 14. Reference and actual paths of the Sawyer robot and updated path parameter in every iteration under the GE-based path learning method. (a) reference and actual paths of the robot (b) updated path parameter and performance index value.

with adaptive laws [40], [41] and B-spline interpolation to extend their applications and make them effective for paths with uncertain geometric information.

Appendix 1

The performance index with the path parameter vector in the $(j+1)$ th iteration can be rewritten as

$$\begin{aligned} W(\boldsymbol{\sigma}_{j+1}) &= W(\boldsymbol{\sigma}_j) + (W(\boldsymbol{\sigma}_{j+1}) - W(\boldsymbol{\sigma}_j)) \\ &= W(\boldsymbol{\sigma}_j) + \left(\nabla W(\bar{\boldsymbol{\sigma}}_j)\right)^{\mathrm{T}} (\boldsymbol{\sigma}_{j+1} - \boldsymbol{\sigma}_j). \end{aligned} \quad (65)$$

Substituting Eq. (26) into Eq. (65), we have

$$W(\boldsymbol{\sigma}_{j+1}) = \left(1 - \left(\nabla W(\bar{\boldsymbol{\sigma}}_j)\right)^{\mathrm{T}} \boldsymbol{\chi}_j\right) W(\boldsymbol{\sigma}_j). \quad (66)$$

As far as the magnitude in Eq. (66) satisfies $\left|1 - \left(\nabla W(\bar{\boldsymbol{\sigma}}_j)\right)^{\mathrm{T}} \boldsymbol{\chi}_j\right| < 1$, i.e., $\left|1 - \boldsymbol{L}_j^{\mathrm{T}} \boldsymbol{\chi}_j\right| \leq \rho < 1$ in Eq. (27), there is a contractive mapping between $W(\boldsymbol{\sigma}_{j+1})$ and $W(\boldsymbol{\sigma}_j)$ such that $W(\boldsymbol{\sigma}_j)$ decreases over iterations and converges under the iterative learning law in Eq. (26) as iteration number j increases.

Appendix 2

According to Assumption 1 and Eq. (36), it follows that

$$\boldsymbol{\zeta}_j - \boldsymbol{\zeta}_{j-1} = \frac{\boldsymbol{\vartheta}_j \boldsymbol{\theta}(\boldsymbol{\sigma}_j) \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j)}{r + \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j) \boldsymbol{\vartheta}_j \boldsymbol{\theta}(\boldsymbol{\sigma}_j)} (\boldsymbol{\zeta}^* - \boldsymbol{\zeta}_{j-1}). \quad (67)$$

Then Eq. (67) is rewritten as

$$\boldsymbol{\zeta}_j - \boldsymbol{\zeta}^* + \boldsymbol{\zeta}^* - \boldsymbol{\zeta}_{j-1} = \frac{\boldsymbol{\vartheta}_j \boldsymbol{\theta}(\boldsymbol{\sigma}_j) \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j)}{r + \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j) \boldsymbol{\vartheta}_j \boldsymbol{\theta}(\boldsymbol{\sigma}_j)} (\boldsymbol{\zeta}^* - \boldsymbol{\zeta}_{j-1}). \quad (68)$$

We define

$$\triangle \boldsymbol{\zeta}_j = \boldsymbol{\zeta}_j - \boldsymbol{\zeta}^* \quad (69)$$

and then substituting Eq. (69) into Eq. (68) yields

$$\triangle \boldsymbol{\zeta}_j = \left(\boldsymbol{I} - \frac{\boldsymbol{\vartheta}_j \boldsymbol{\theta}(\boldsymbol{\sigma}_j) \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j)}{r + \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j) \boldsymbol{\vartheta}_j \boldsymbol{\theta}(\boldsymbol{\sigma}_j)}\right) \triangle \boldsymbol{\zeta}_{j-1} \quad (70)$$

where $\boldsymbol{I}$ is an identity matrix with proper dimensions.

Since the following inequality holds

$$\left\| \boldsymbol{I} - \frac{\boldsymbol{\vartheta}_j \boldsymbol{\theta}(\boldsymbol{\sigma}_j) \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j)}{r + \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{\sigma}_j) \boldsymbol{\vartheta}_j \boldsymbol{\theta}(\boldsymbol{\sigma}_j)} \right\| < 1, \quad (71)$$

the contractive mapping exists between $\triangle \boldsymbol{\zeta}_j$ and $\triangle \boldsymbol{\zeta}_{j-1}$ such that

$$\lim_{j \to \infty} \|\boldsymbol{\zeta}_j - \boldsymbol{\zeta}^*\| = 0 \quad (72)$$

$$\lim_{j \to \infty} \left| \hat{W}(\boldsymbol{\sigma}_j) - W(\boldsymbol{\sigma}_j) \right| = 0 \quad (73)$$

which indicates from the perspective of continuity that

$$\lim_{j \to \infty} \left\| \nabla \hat{W}(\boldsymbol{\sigma}_j) - \nabla W(\boldsymbol{\sigma}_j) \right\| = 0. \quad (74)$$

Define

$$\Delta W = W(\boldsymbol{\sigma}_{j+1}) - W(\boldsymbol{\sigma}_j). \quad (75)$$

According to Assumption 2, it is obvious that

$$\begin{aligned}\Delta W &= \boldsymbol{\nabla} W^{\mathrm{T}}\left(\boldsymbol{\sigma}_j^*\right)\left(\boldsymbol{\sigma}_{j+1}-\boldsymbol{\sigma}_j\right)\\ &= \boldsymbol{\nabla} W^{\mathrm{T}}\left(\boldsymbol{\sigma}_j\right)\left(\boldsymbol{\sigma}_{j+1}-\boldsymbol{\sigma}_j\right)\\ &+\left(\boldsymbol{\nabla} W^{\mathrm{T}}\left(\boldsymbol{\sigma}_j^*\right)-\boldsymbol{\nabla} W^{\mathrm{T}}\left(\boldsymbol{\sigma}_j\right)\right)\left(\boldsymbol{\sigma}_{j+1}-\boldsymbol{\sigma}_j\right)\\ &\leq \boldsymbol{\nabla} W^{\mathrm{T}}\left(\boldsymbol{\sigma}_j\right)\left(\boldsymbol{\sigma}_{j+1}-\boldsymbol{\sigma}_j\right)\\ &+\left\|\boldsymbol{\nabla} W^{\mathrm{T}}\left(\boldsymbol{\sigma}_j^*\right)-\boldsymbol{\nabla} W^{\mathrm{T}}\left(\boldsymbol{\sigma}_j\right)\right\|\left\|\boldsymbol{\sigma}_{j+1}-\boldsymbol{\sigma}_j\right\|\\ &\leq \boldsymbol{\nabla} W^{\mathrm{T}}\left(\boldsymbol{\sigma}_j\right)\left(\boldsymbol{\sigma}_{j+1}-\boldsymbol{\sigma}_j\right)+\alpha\left\|\boldsymbol{\sigma}_{j+1}-\boldsymbol{\sigma}_j\right\|^2. \end{aligned} \tag{76}$$

Substituting Eq. (33) into inequality (76), we have

$$\Delta W \leq -\boldsymbol{\nabla} W^{\mathrm{T}}\left(\boldsymbol{\sigma}_j\right)\boldsymbol{K_J}\boldsymbol{\nabla}\hat{W}\left(\boldsymbol{\sigma}_j\right)+\alpha\left\|\boldsymbol{K_J}\nabla\hat{W}\left(\boldsymbol{\sigma}_j\right)\right\|^2. \tag{77}$$

From inequality (77), it is known that the following inequality holds for sufficiently large j

$$\begin{aligned}\Delta W &\leq -\boldsymbol{\nabla} W^{\mathrm{T}}\left(\boldsymbol{\sigma}_j\right)\boldsymbol{K_J}\boldsymbol{\nabla} W\left(\boldsymbol{\sigma}_j\right)+\alpha\left\|\boldsymbol{K_J}\boldsymbol{\nabla} W\left(\boldsymbol{\sigma}_j\right)\right\|^2\\ &\leq -\psi_{\min}\left(\boldsymbol{K_J}\right)\left\|\boldsymbol{\nabla} W\left(\boldsymbol{\sigma}_j\right)\right\|^2+\alpha\left\|\boldsymbol{K_J}\boldsymbol{\nabla} W\left(\boldsymbol{\sigma}_j\right)\right\|^2\\ &\leq -\left(\psi_{\min}\left(\boldsymbol{K_J}\right)-\alpha\left\|\boldsymbol{K_J}\right\|^2\right)\left\|\boldsymbol{\nabla} W\left(\boldsymbol{\sigma}_j\right)\right\|^2. \end{aligned} \tag{78}$$

By observing inequality (78), it follows that when the condition inequality $\psi_{\min}\left(\boldsymbol{K_J}\right)-\alpha\left\|\boldsymbol{K_J}\right\|^2>0$ holds, $\Delta W<0$ holds, which means that $W\left(\boldsymbol{\sigma}_j\right)$ decreases over iterations and converges with the iterative adaptation law in Eq. (33).

APPENDIX 3

According to Eq. (21), we can obtain the following expression of the closed-system in the jth iteration

$$\boldsymbol{M_d}\ddot{\boldsymbol{e}}_j\left(t\right)+\boldsymbol{k_d}\dot{\boldsymbol{e}}_j\left(t\right)+\boldsymbol{k_p}\boldsymbol{e}_j\left(t\right)=\boldsymbol{M_d}\boldsymbol{M}^{-1}\left(t\right)\boldsymbol{F}_{\boldsymbol{h},j}\left(t\right). \tag{79}$$

Suppose the interaction force $\boldsymbol{F}_{\boldsymbol{h},j}\left(t\right)$ is bounded due to limited strength of human arm, and then $\boldsymbol{e}_j\left(t\right)$ is bounded. As demonstrated in Section III-A and Section III-B, the performance index function in Eq. (43) can be iteratively decreased and minimized to zero, which means $\lim_{j\to\infty}\boldsymbol{F}_{\boldsymbol{h},j}\left(t\right)=0$ holds.

The closed-loop system when $j\to\infty$ can then be rewritten as

$$\boldsymbol{M_d}\ddot{\boldsymbol{e}}_m\left(t\right)+\boldsymbol{k_d}\dot{\boldsymbol{e}}_m\left(t\right)+\boldsymbol{k_p}\boldsymbol{e}_m\left(t\right)=0 \tag{80}$$

where $m=j\to\infty$. It is trivial to prove that $\boldsymbol{e}_m\to 0$ [42].

REFERENCES

[1] C. Yang, C. Zeng, P. Liang, Z. Li, R. Li, and C.-Y. Su, "Interface design of a physical human-robot interaction system for human impedance adaptive skill transfer," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 329–340, 2017.

[2] W. He, C. Xue, X. Yu, Z. Li, and C. Yang, "Admittance-based controller design for physical humancrobot interaction in the constrained task space," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1937–1949, 2020.

[3] X. Wu, Z. Li, Z. Kan, and H. Gao, "Reference Trajectory Reshaping Optimization and Control of Robotic Exoskeletons for Human-Robot Co-Manipulation," *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3740–3751, 2020.

[4] M. R. Fraune, S. Sherrin, S. Šabanović, and E. R. Smith, "Is human-robot interaction more competitive between groups than between individuals?" in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2019, pp. 104–113.

[5] E. Noohi, M. Žefran, and J. L. Patton, "A model for human–human collaborative object manipulation and its application to human–robot interaction," *IEEE transactions on robotics*, vol. 32, no. 4, pp. 880–896, 2016.

[6] F. Vicentini, M. Askarpour, M. G. Rossi, and D. Mandrioli, "Safety assessment of collaborative robotics through automated formal verification," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 42–61, 2019.

[7] L. Roveda, S. Haghshenas, M. Caimmi, N. Pedrocchi, and L. M. Tosatti, L. M. , "Assisting Operators in Heavy Industrial Tasks: On the Design of an Optimized Cooperative Impedance Fuzzy-Controller With Embedded Safety Rules," *Frontiers in Robotics and AI*, vol. 6, 2019.

[8] Y. Li and S. S. Ge, "Human-robot collaboration based on motion intention estimation," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2013.

[9] L. Roveda, J. Maskani, P. Franceschi, A. Arash, and N. Pedrocchi, "Model-Based Reinforcement Learning Variable Impedance Control for Human-Robot Collaboration," *Journal of Intelligent & Robotic Systems*, 2020.

[10] N. Delson and H. West, "Robot programming by human demonstration: The use of human inconsistency in improving 3d robot trajectories," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, vol. 2. IEEE, 1994, pp. 1248–1255.

[11] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras, Carme, "Learning physical collaborative robot behaviors from human demonstrations," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 513–527, 2016.

[12] J. Silvério, S. Calinon, L. Rozo, and D. G. Caldwell, "Learning task priorities from demonstrations," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 78–94, 2018.

[13] Z. Li, J. Liu, Z. Huang, Y. Peng, H. Pu, and L. Ding, "Adaptive impedance control of human–robot cooperation using reinforcement learning," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 8013–8022, 2017.

[14] A. H. Qureshi, Y. Nakamura, Y. Yoshikawa, and H. Ishiguro, "Intrinsically motivated reinforcement learning for human–robot interaction in the real-world," *Neural Networks*, vol. 107, pp. 23–33, 2018.

[15] M. Awais and D. Henrich, "Human-robot interaction in an unknown human intention scenario," in *2013 11th International Conference on Frontiers of Information Technology*. IEEE, 2013, pp. 89–94.

[16] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6015–6022.

[17] L. Roveda, M. Forgione, and D. Piga , "Robot Control Parameters Auto-Tuning in Trajectory Tracking Applications," *Control Engineering Practice*, vol. 101, 2020.

[18] L. Roveda, M. Magni, M. Cantoni, D. Piga, and G. Bucca , "Human-robot collaboration in sensorless assembly task learning enhanced by uncertainties adaptation via Bayesian Optimization," *Robotics and Autonomous Systems*, vol. 136, pp. 103 711, 2021.

[19] L. Rozo , "Interactive Trajectory Adaptation through Force-guided Bayesian Optimization," *arXiv preprint arXiv:1908.07263*, 2019.

[20] S. Mandra, K. Galkowski, A. Rauh, H. Aschemann, and E. Rogers, "Iterative learning control for a class of multivariable distributed systems with experimental validation," *IEEE Transactions on Control Systems Technology*, pp. 1–12, 2020.

[21] W. He, T. Meng, X. He, and C. Sun, "Iterative learning control for a flapping wing micro aerial vehicle under distributed disturbances," *IEEE Transactions on Cybernetics*, vol. 49, no. 4, pp. 1524–1535, 2019.

[22] J. Zhang, C. C. Cheah, and S. H. Collins, "Experimental comparison of torque control methods on an ankle exoskeleton during human walking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5584–5589.

[23] D. P. Losey and M. K. O'Malley, "Learning the correct robot trajectory in real-time from physical human interactions," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 9, no. 1, pp. 1–19, 2019.

[24] Y. Li and S. S. Ge, "Force tracking control for motion synchronization in human-robot collaboration," *Robotica*, vol. 34, no. 06, pp. 1260–1281, 2014.

[25] R. B. Gillespie, J. E. Colgate, and M. A. Peshkin, "A general framework for cobot control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 391–401, 2001.

[26] Q. Zeng, C. L. Teo, B. Rebsamen, and E. Burdet, "A collaborative wheelchair system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, no. 2, pp. 161–170, 2008.

[27] Z. Zhang, Y. Lin, S. Li, Y. Li, Z. Yu, and Y. Luo, "Tricriteria optimization-coordination motion of dual-redundant-robot manipulators

for complex path planning," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1345–1357, 2018.

[28] D. Guo, F. Xu, and L. Yan, "New pseudoinverse-based path-planning scheme with pid characteristic for redundant robot manipulators in the presence of noise," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 6, pp. 2008–2019, 2017.

[29] Y. Zhao, Z. Zheng, X. Zhang, and Y. Liu, "Q learning algorithm based uav path learning and obstacle avoidence approach," in *2017 36th Chinese Control Conference (CCC)*. IEEE, 2017, pp. 3397–3402.

[30] M. K. Cobb, K. Barton, H. Fathy, and C. Vermillion, "Iterative learning-based path optimization for repetitive path planning, with application to 3-d crosswind flight of airborne wind energy systems," *IEEE Transactions on Control Systems Technology*, 2019.

[31] A. G. Feldman, "Once more on the equilibrium-point hypothesis (λ model) for motor control," *Journal of motor behavior*, vol. 18, no. 1, pp. 17–54, 1986.

[32] B. Siciliano and V. Luigi, "Robot force control," *Springer Science & Business Media*, vol. 540, 2012.

[33] Y. Lou, H. Meng, J. Yang, Z. Li, J. Gao, and X. Chen, "Task polar coordinate frame-based contouring control of biaxial systems," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3490–3501, 2013.

[34] R. B. Gillespie, J. E. Colgate, and M. A. Peshkin, "A general framework for cobot control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 391–401, 2001.

[35] R. W. Hogg, A. L. Rankin, S. I. Roumeliotis, M. C. McHenry, D. M. Helmick, C. F. Bergh, and L. Matthies, "Algorithms and sensors for small robot path following," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 4. IEEE, 2002, pp. 3850–3857.

[36] J.-X. Xu, D. Huang, and S. Pindi, "Optimal tuning of pid parameters using iterative learning approach," *SICE Journal of Control, Measurement, and System Integration*, vol. 1, no. 2, pp. 143–154, 2008.

[37] J. X. Xu, and Y. Tan, "Linear and nonlinear iterative learning control," in *Springer*, 2003.

[38] R. Salomon, "Evolutionary algorithms and gradient search: similarities and differences," *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 2, pp. 45–55, 1998.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *the 3rd International Conference for Learning Representations*, 2015, pp. 1–15.

[40] W. He, X. Mu, L. Zhang, and Y. Zou, "Modeling and trajectory tracking control for flapping-wing micro aerial vehicles," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 1, pp. 148–156, 2021.

[41] Z. Liu, Z. Han, Z. Zhao, and W. He, "Modeling and adaptive control for a spatial flexible spacecraft with unknown actuator failures," *Sciece China. Information Sciences*, vol. 64, 10 2020.

[42] T. Kuc, K. Nam, and J. S. Lee, "An iterative learning control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 835–842, 1991.